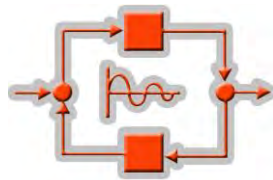




The International Congress for global Science and Technology



ICGST International Journal on Automatic Control & System Engineering (ACSE)

**Volume (17), Issue (II)
December 2017**

**www.icgst.com
www.icgst-amc.com
www.icgst-ees.com**

**© ICGST LLC, 2017
Delaware, USA**

ACSE Journal

ISSN: Print 1687-4811

ISSN Online 1687-482X

ISSN CD-ROM 1687-4838

© ICGST LLC, 2017, Delaware, USA

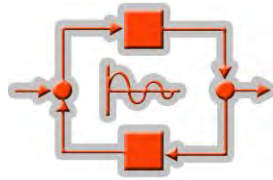
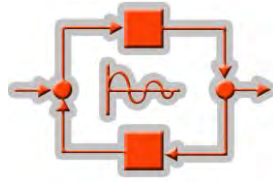


Table of Contents

Papers	Pages
P1130945914, author="Alaa Sheta and Mohammad Salamah", title="An Innovative NXT Mobile Robot System for Landmine Detection",	1--8
P1111733578, author="Prof. Dr. Kamen M. Yanev", title="Advanced D-Partitioning Stability Analysis of Digital Control Systems with Multivariable Parameters",	9--19
P1111748593, author="Ali Alqudaihi and Mohamed Zohdy and Hoda Abdel-Aty-Zohdy", title="Multi-Sensor Fusion Indoor Security Based on Interval Probability, Extended Kalman Filter and Fuzzy Rules Pruning",	21--28
P1111744588, author="Kamal ElDahshan and Afaf Abd Elkader and Nermeen Ghazy", title="Round Robin based Scheduling Algorithms, A Comparative Study",	29--42



**ICGST International Journal on Automatic Control & System Engineering -
(ACSE)**

**A Publication of the International Congress for global Science and Technology -
(ICGST)**

ICGST Editor in Chief: Dr. rer. nat. Ashraf Aboshosha

www.icgst.com, www.icgst-amc.com, www.icgst-ees.com

editor@icgst.com

Abstracts

P1130945914,

author="Alaa Sheta and Mohammad Salamah",

title="An Innovative NXT Mobile Robot System for Landmine Detection",

Abstract: The current landmine detection technologies suffer many challenges on solving the humanitarian demining problem. Until now, detection and clearance in humanitarian demining very often relies on manual methods such as metal detector, as a primary procedure. The aim of this paper is to propose a multistage image processing methodology to detect the location of landmines in IR images. The methodology count on the use of morphological image processing to detect the target mines. A database of images provided by the Royal Military Academy of Belgium was used to develop our results. The proposed MIP methodology was integrated with an NXT mobile robot, a smart N80 mobile phone and PC Server to test the developed software on a grid based environment. It was found that the proposed methodology and integrated system is very promising.

P1111733578,

author="Prof. Dr. Kamen M. Yanev",

title="Advanced D-Partitioning Stability Analysis of Digital Control Systems with Multivariable Parameters",

Abstract: This paper contributes in further advancement of the method of D-Partitioning stability analysis applied to digital control system with multivariable parameters. It also explores the effects of simultaneous system uncertainties by determining graphically regions of stability in the space of the system's parameters. The Euler's approximation is taken into account and the Bilinear Tustin Transform method is employed as one of the most precise facilities for systems discretization. The interaction between the variable parameters will bring a new light in the graphical solution of the problem of control system stability, where the analysis of control systems with two and three simultaneously variable parameters is taken into consideration. The proposed research is a continuation and further development of the already published author's work on the advancement of the method of the D-Partitioning stability analysis of digital control systems with variable parameters. The suggested implementation for system's stability analysis is fundamental for the further advancement of control theory in this area.

P1111748593,

author="Ali Alqudaihi and Mohamed Zohdy and Hoda Abdel-Aty-Zohdy",

title="Multi-Sensor Fusion Indoor Security Based on Interval Probability, Extended Kalman Filter and Fuzzy Rules Pruning",

Abstract: In this paper, noisy heterogeneous multi-sensors, namely Infrared, Active Sonar and Pressure sensors are used for indoor security monitoring. A new approach of enhancing the measurement estimation applying an interval probability to the Extended Kalman Filter process and measurement covariances. A total of four different combinations of the covariances were examined by the addition of upper and lower distributions bounds which helped to find the finest combination that keeps the covariance matrices symmetric and positive definite. In addition, Fuzzy-Logic I rules pruning methods and their applications in security monitoring systems and decision-making rules are addressed and discussed.

P1111744588,

author="Kamal ElDahshan and Afaf Abd Elkader and Nermeen Ghazy",

title="Round Robin based Scheduling Algorithms, A Comparative Study",

Abstract: Scheduling is the process of allocating processes to the CPU in order to optimize some objective function. There are many algorithms used to schedule processes. The Round Robin (RR) CPU scheduling algorithm is one of these algorithms which is effective in time sharing and real time operating systems. It gives reasonable response time. But it suffers from several disadvantages such as high turnaround time, high waiting time and many context switches. There are large numbers of algorithms proposed to enhance the standard Round Robin algorithm. In this paper we present a survey with results analysis that conclude recommendations for an Enriched Round Robin algorithm that ameliorates the performance of average waiting time and average turnaround time.



An Innovative NXT Mobile Robot System for Landmine Detection

Alaa Sheta and Mohammad Salamah

Information Technology Department, Al-Balqa Applied University, Salt, Jordan
asheta2@yahoo.com, m.omar82@gmail.com

Abstract

The current landmine detection technologies suffer many challenges on solving the humanitarian demining problem. Until now, detection and clearance in humanitarian demining very often relies on manual methods such as metal detector, as a primary procedure. The aim of this paper is to propose a multistage image processing methodology to detect the location of landmines in IR images. The methodology count on the use of morphological image processing to detect the target mines. A database of images provided by the Royal Military Academy of Belgium was used to develop our results. The proposed MIP methodology was integrated with an NXT mobile robot, a smart N80 mobile phone and PC Server to test the developed software on a grid based environment. It was found that the proposed methodology and integrated system is very promising.

Keywords: Landmine detection, Morphological Operation, NXT Mobile Robot

1. Introduction

The development of demining technologies is a quite difficult and costly task. This is because of the wide range of terrains, which could be rocky, desert, beaches, muddy, river, forest, etc [1], [2]. It is important to mention that the climate conditions affect very much the available sensor technologies for landmine detection such as the Infrared (IR) technology [3]. IR technology is mostly affected by hot, humid, rainy, cold, windy conditions [4].

Humanitarian demining is a challenging process especially for the third world countries [5], [6]. To clear an area from mines, it requires that the removal process achieves a success ratio of 100%. The land should be free of mines with a high reliability and safety for human. Failure to do that will cost people life. The amount of time it takes to clear an area is less important than the safety of the clearance. Safety is with the highest importance. In military situation sometimes casualties are acceptable. This is not true in the case of civilians. We must also have a mechanism which results in no risks to those handling the demining task.

Thermal infrared cameras often used to detect landmines based on the assumption that landmines have different thermal properties than their surrounding soil. When an area is heated due to solar radiation, the landmines will warm up quicker than the soil. The soil above the landmine also warms up quickly.

The influence of the landmine on the temperature of the soil at surface level relates to the burial depth. This situation, where landmines give relatively warm spots at the surface cause what is called *positive contrast*. When an area is cooled, for instance some time after sunset, the situation is reversed. Landmines will cause cold spots at the surface and cause what is called *negative contrast*.

2. Promising Sensor Technology

In April 2001, a report was written to the Chairman, Subcommittee on Military Research and Development, Committee on Armed Services, House of Representatives entitled, "Landmine Detection DoD's Research Program Needs a Comprehensive Evaluation Strategy". The committee objective was to evaluate the DoD's strategy for identifying the most promising land mine detection technologies. The report stated:

Improving DoD's land mine detection capability is a challenging technological issue. Because of the threat that land mines pose to U.S. armed forces, you requested that we assess the abilities of competing technological options to address DoD's mission needs for land mine detection. Specifically, our objectives were to determine whether DoD (1) employs an effective strategy for identifying and evaluating the most promising land mine detection technologies and (2) is investing in the most promising technologies to fully address mission needs.

The DoD search results show that: there are nine potentially promising technologies are funded by the DoD. They include: Acoustics/Seismic, Biosensors, Infrared/Multi-hyperspectral, LIDAR, Radar, Electromagnetic radiography, Passive millimeter wave, Terahertz imaging and X-ray fluorescence. These nine technologies proved to work in diverse environmental conditions with limited constraints related to soil types and water in soil.

Exploration of new technologies are being investigated to improve the reliability and speedup the detection operation, some of these technologies are Electromagnetic Induction Metal detectors (EMI) [7], Infrared Imaging, Ground-Penetrating Radar (GPR) [8], Acoustics, Thermal Neutron Activation (TNA), Photoacoustic Spectroscopy, Nuclear Quadrupole Resonance (NQR), X-ray Tomography, Neutron Back-scattering, Biosensors, Commercial sniffers, etc.



2.1 Infrared (IR) Imaging

IR radiation constitutes a portion of the Electromagnetic Spectrum (EM) domain. The frequency band of the EM waves is between 100 MHz and 100 GHz [9]. This is a fairly high frequency band. IR spectrum span in the wavelengths domain of 75 μ m and 1mm [9]. It is known that EM radiation produces heat. IR radiation can respond to the heat based on the type of material. Using EM to get the emission response represents a good source of infrared radiation.

IR has been used widely for handling the landmine detection problem. One drawback of IR imaging is that IR relies highly on the environment at the moment of measurement and the type of soil in the background. Infrared methods detect variations in electromagnetic radiation reflected or emitted by either the mines surface or the soil and vegetation located above the buried landmines [1].

IR imaging requires sensitive cameras with sufficient resolution. This technology measures landmines at a maximum burial depth of 10-15 cm [9]. In general the strength of infrared signatures is independent of the landmine class, and varies through the day. It is a day to day dependent process depending on environmental conditions such as solar loading, precipitation, and wind.

3. The Proposed MIP Methodology

Our objective is to detect the location of landmines in Infrared images using a Multistage Image Processing (MIP) methodology (see Figure 1) and integrated mobile robot system. Number of image processing methodologies for landmine detection were proposed in [10], [11], [12]. The methodology starts by converting color image to gray scale. In the second stage, we select a suitable structuring element for performing morphological operation. We make image opening and closing on the gray scale image. In stage three, we check on the existence of objects in the image by selecting a suitable threshold level using our proposed methodology. By the end of this stage, we will be having a binary image with the detected object. In stage four, we label the connected components and use the Blob analysis technique to measure the properties for each labeled region [13].

3.1 Stage 1: Morphological Operations

Mathematical Morphology is a powerful tool for extracting useful components from an image. Morphological image processing technique was provided in [14] at the Ecole des Mines in Paris. The technique was introduced as a set-theoretic method for image analysis which could help in extracting geometrical structures from an image. Using morphology, we can provide boundaries of objects, their skeletons, and their convex hulls [15]. It was also found useful in pre- and post-processing phases of image processing, especially in edge thinning and pruning [16], [17].

Morphological image processing is a collection of techniques for digital image processing based on mathematical morphology [18]. Morphology Operations are useful in the representation and description of region shape since these techniques rely only on the relative ordering of pixel values, not on their numerical values, they are especially suited to the processing of binary images and gray scale images. In this stage, it is important to select an appropriate structure element for the opening and closing processes. It will significantly help in separating the object from the background. Since we are searching for a landmine shape, we decided to use a structuring element S of a disk-shape [19].

3.2 Stage 2: Thresholding

Threshold is an important technique for image segmentation which helps to identify or extract a target from an image. Threshold can significantly help in achieving this goal if a proper threshold-level is chosen. The appropriate level helps to classify the foreground and the background. Pixels that have a value of intensity level greater than the threshold level will be classified as a foreground. Increasing the threshold level reduces the number of false alarms, but also leads to a lower detection rate [20].

In the past, number of threshold techniques was proposed.

In [21], image threshold method which combines various multi-scale and morphological features, including texture, shape and edge filtering by using genetic programming was presented. A faster version of Otsu's method was proposed for improving the efficiency of computation for the optimal thresholds of an image [22]. A survey over image threshold techniques and quantitative performance evaluation was presented in [23], [24] and novel parametric and global image histogram threshold method was presented in [25].

The proposed threshold technique can be summarized in the following steps (see Figure 2):

- Obtain the image histogram.
- Locate the points of minimum gray scale level and consider it as local minimum. In the case of an image with one target (see Figure 2), the global minimum between the two peaks will be selected as the thresholding level. For an image with multiple minimum points (see Figure 3), we test all minimum levels till we locate our target.
- Remove all objects which have fewer than 100 pixels.
- Check if there is an object in the image, if not we tests another threshold level.

In Figure 3, we can see that the local minimum levels are located at the values 104, 122, 131, 153 and 159. Processing the image with the given threshold levels, we get the images provided in Figure 4 with landmine detected at level 159.

3.3 Stage 3: Feature Extraction

The binary large object (blob) is an area of connecting pixels with the same logical state. All pixels in an image that



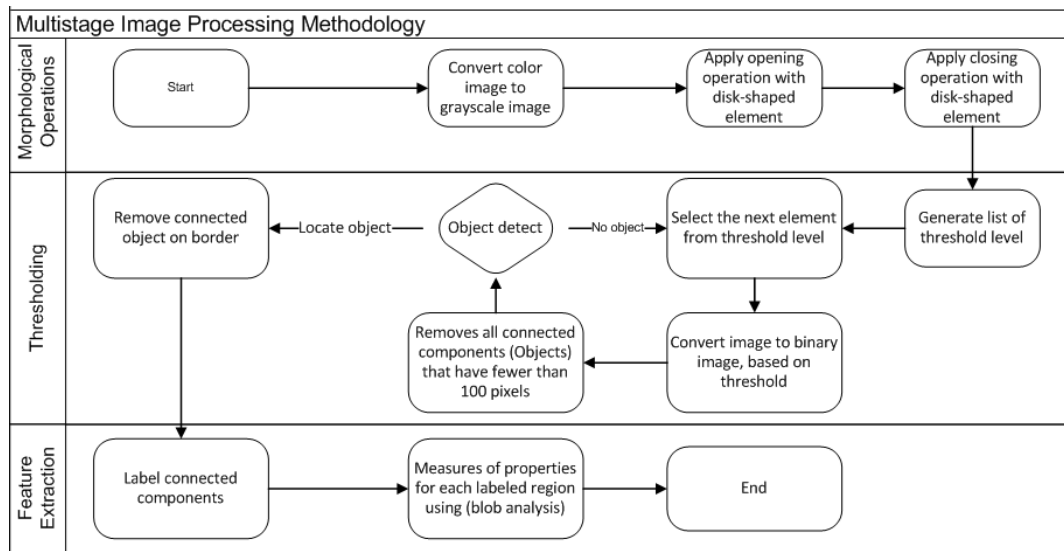


Fig. 1: A flow chart of the Multistage Image Processing (MIP) methodology

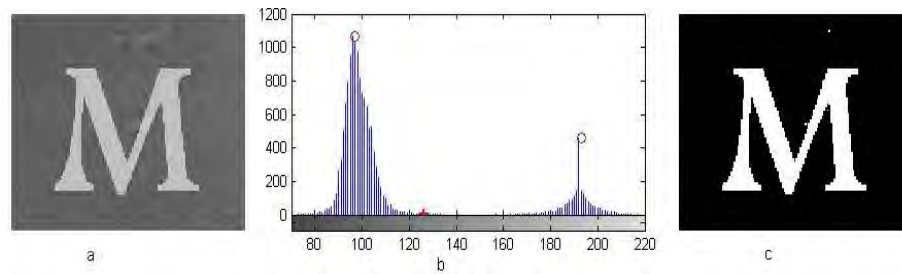


Fig. 2: (a) Original image with one object (b) Image histogram (c) Processed image with the object detected

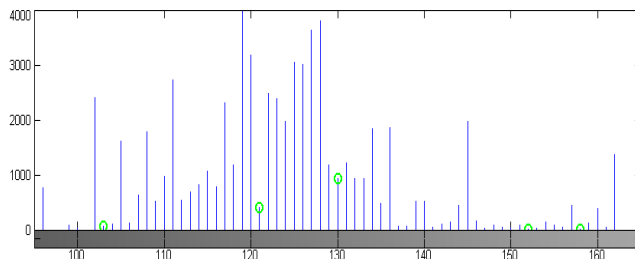


Fig. 3: Image histogram with multiple local minimum at levels 104, 122, 131, 153 and 159 of an IR image with landmine

belong to a blob are in a foreground class. All other pixels are in a background class. In a binary image, pixels in the background have values equal to zero while every nonzero pixel is part of a binary object [26]. Blob analysis is used to get 1) edges locations; 2) center location; 3) number of pixels in an object [13].

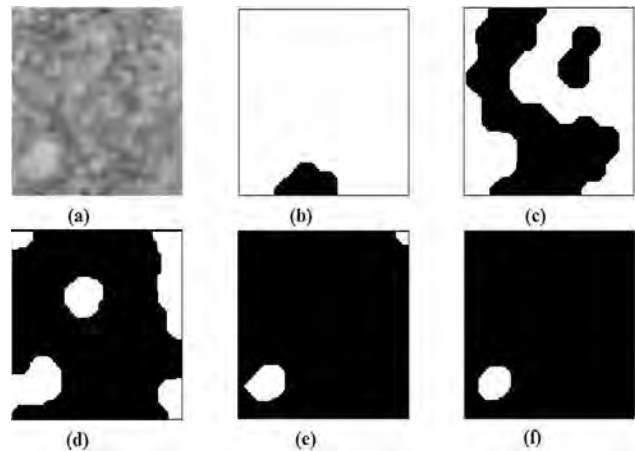


Fig. 4: (a) Original image. Processed IR image with threshold levels (b)104 (c)122 (d)131 (e)153 (f)159



4. Difficulties of Landmine Detection Using Robotics

A cheap and fast solutions based mobile robot system is urgently needed especially for landmine applications. Mine-seeking robots are still in the premature development stages. Currently, mobile robot systems are not flexible for their intended missions or can provide cost effective solution for mine detection and removal. This is why there is a need for newly designed robotics hardware and software systems which can be attractive to their missions. Some of the difficulties associated with using robotics for landmine detection was described in [2], [27] and can be summarized as follows:

- Most robotic equipment is extremely expensive to build, expensive to run and expensive to maintain.
- The robotics technology in the countries where landmine is buried is very low. The complexity of the robots sometimes makes it difficult to excite the end users about it.
- Many problems arise when we deal with sensitive applications such as the landmine problem. This is because most of laboratory robots are not qualified for real detection and clearing environment (i.e. vegetation, sand and mud).
- A multi-robot system or group of robots could provide a fast and efficient solution to the problem although the development of a control system to manage and activity for such systems is a complex task [28].

5. Basic Requirement of a Robot System

In [27] author stated that:

Before applying robotics technology for the mine clearance process, it is necessary to specify the basic requirements for a robot to have in order to achieve a better performance. These requirements include mechanisms, algorithms, functions and use.

These requirements were described in [27]. It can be itemized as follows:

- The robot should have a light weight not to activate a landmine.
- The robot should have a simple design, flexible motion and highly reliable to accidents.
- The robot should be cheap to run, small, lightweight, and portable.
- The robot design should have adequate number of sensors to overcome obstacles and select a path.
- The robot should be able to recover problems which could happen during navigation and probing.
- The robot should be able to work in environment such as water, sand, temperature and humidity [29].
- The mechanical design of the robot should be simple for maintenance and repair.

- The robot high technology parts should be well protected.
- The robot power supply should enable the robot to operate for long period of time.
- The robot should have a remote control facility for tele-operation in landmine fields.

6. The Proposed Mobile Robot System

The demining process takes long time to accomplish and a minefield is one of the most dangerous regions. This paper proposes an integrated system which can speed up the process of mine detection and reduce risk than traditional manual methods for detection.

The proposed system relies on moving an NXT mobile robot to scan a minefield based on a developed grid environment of the mine field and detecting mines by using appropriate sensors attached to the mobile robot. The system developed a map of the area discovered, showing positions of located mines. The Proposed system consists of three devices as shown in Figure 5.

- 1) A PC Server is a Pentium 4 computer with a 512 RAM and support Wi-Fi connection. The PC Server is the brain for the whole system since it has the monitoring and the control system.
- 2) The LEGO NXT is the mobile robot which will be used to execute the mission. The NXT will hold the N80 mobile phone and navigate in the minefield.
- 3) N80 is a smart mobile phone device. N80 has an ARM-9 processor with clock rate of 220 MHz CPU, 18 MB RAM and Symbian OS v9.1¹. N80 include 3.0 Mega pixels camera and support for different type of wireless communications. They include (1) 3G network (384 kbps), (2) EDGE (286.8 kbps) (3) Bluetooth 2.0 and 4) WLAN 802.11b/g (See Figure 6).

The proposed system assumes we can have an IR camera to be carried on the mobile robot. We suggest the use of the 400DX IR camera to be attached to a PDA and used as a mobile wireless IR camera. The developed environment for our experimentation is shown in Figure 7. We considered the environment as a grid of steps each of which with a fixed selected size depending on the size of the mine field. A software system for the MIP was developed to apply the proposed MIP methodology. The user interface of the software is presented in Figure 8. The software was developed using MATLAB programming language.

7. Experimental Results

Our objective is to detect landmine from IR images using a proposed MIP methodology and integrated mobile robot

¹Symbian OS is a proprietary operating system, designed for mobile devices, with associated libraries, user interface frameworks and reference implementations of common tools, produced by Symbian Ltd. It is a descendant of Psion's EPOC and runs exclusively on ARM processors [30]



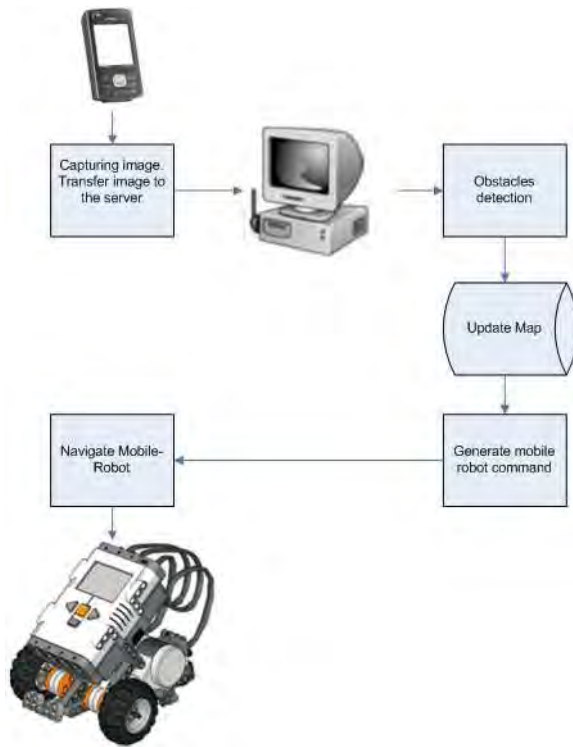


Fig. 5: The proposed mobile robot system

system. As we mentioned, there is normally a difference between the grayscale of the landmine and the grayscale of the soil.

We run our experiment on real images provided by the Royal Military Academy of Belgium, Belgium of MWIR. The MWIR database provides 136 IR images for both *sand* and *gravel* soils [31]. These images were collected at Meerdael (Belgium) minefields on 1st, 2nd and 3rd of April 1998 using mid-wave infrared cameras - AGEMA ($3\mu\text{m}$ - $5\mu\text{m}$).

In Figure 9 and Figure 10, we show the original IR images with landmine, the corresponding histogram of the image and the image processed using the proposed MIP methodology. The two images were taken in a sand soil at the time 13:08 and 22:04, respectively.

We applied our proposed methodology on two images with a gravel soil background. The results of the image processing methodology is presented in Figure 11 and Figure 12. The two images were taken at the time 12:18 and 22:04, respectively. The developed results in the two soil environment was compared by the results provided by [32]. The location of the landmine was verified successfully. The proposed threshold level was selected appropriately. This means that the proposed methodology works well with both the day and evening time images.



Fig. 6: An N80 mobile phone fixed on the NXT robot

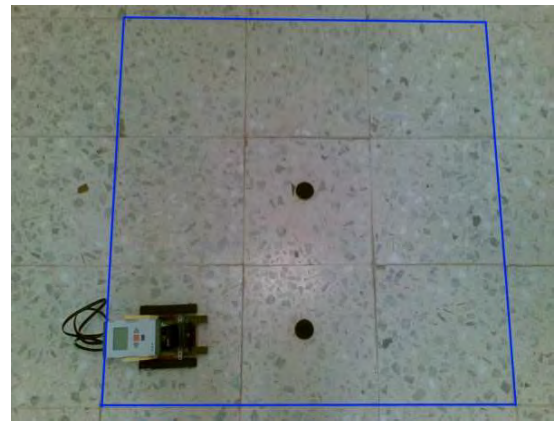
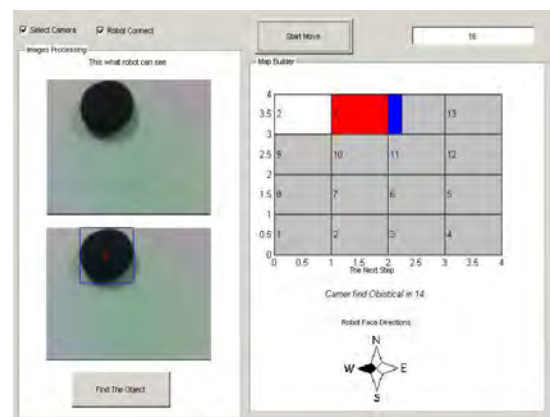


Fig. 7: The environment for the experimentation Fig.



8: The developed software interface



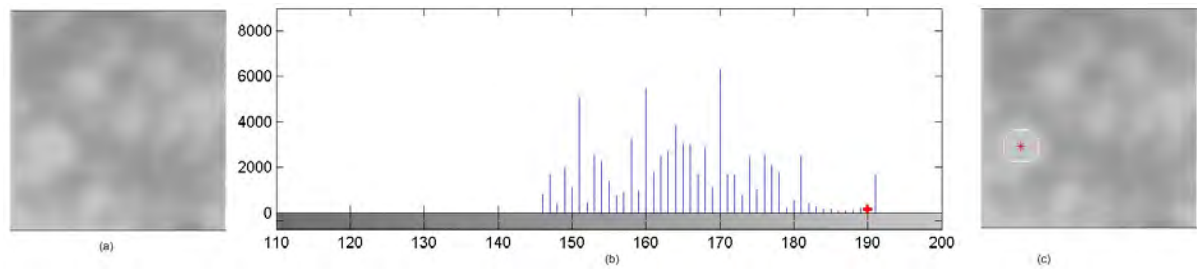


Fig. 9: (a) Original IR image on a sand soil taken at a time 13:08 (b) Image histogram (c) Processed image with mine detected

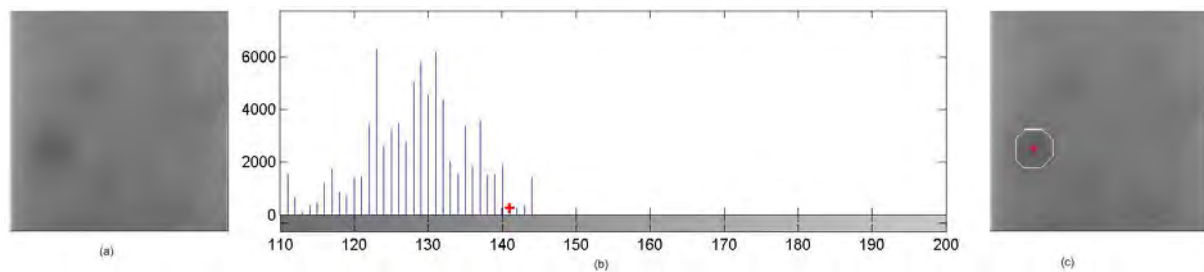


Fig. 10: (a) Original IR image on a sand soil taken at a time 22:04 (b) Image histogram (c) Processed image with mine detected

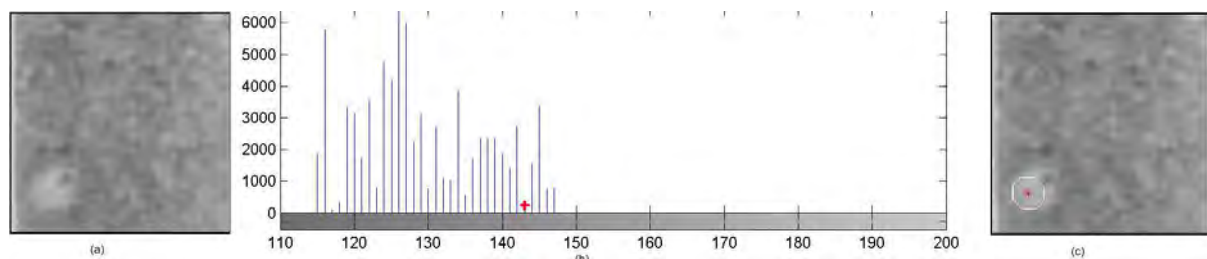


Fig. 11: (a) Original IR image on a gravel soil taken at a time 12:18 (b) Image histogram (c) Processed image with mine detected

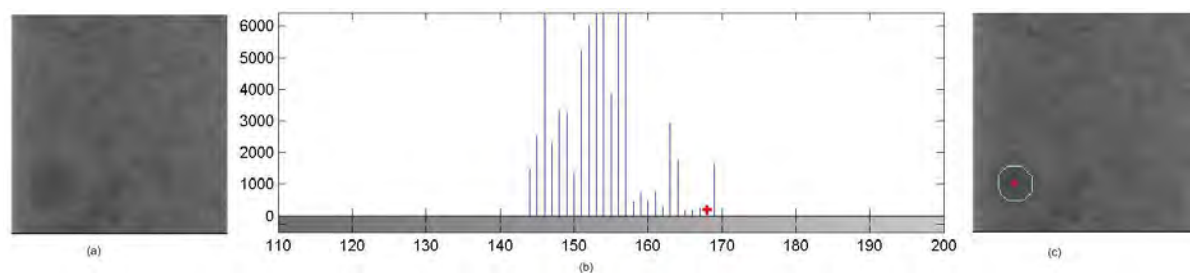


Fig. 12: (a) Original IR image on a gravel soil taken at a time 21:19 (b) Image histogram (c) Processed image with mine detected



8. Conclusions

In this paper, we provided a multi-stage image processing technique which uses morphological image processing to locate landmine in infrared image. The proposed MIP methodology was integrated with a mobile robot system for testing purposes. The process starts by converting color image to gray scale, select suitable structuring element for performing morphological operation, pick up a suitable threshold level to locate the landmine object, produce a binary image with the detected object. Finally, the target is labeled and located. The developed results were verified according to [32] with correct the correct mine locations detected. The developed results show that this system can be expanded with appropriate hardware facility to work in real landmine fields.

References

- [1] S. ho Hong, T. Miller, H. Tobin, B. Borchers, J. M. H. Hendrickx, H. Lensen, P. Schwering, and B. Baertlein, "Impact of soil water content on landmine detection using radar and thermal infrared sensors, detection and remediation technologies for mines and minelike targets VI," in *Detection and Remediation Technologies for Mines and Minelike Targets*, 2001, pp. 409–416.
- [2] M. K. Habib, "Mine clearance techniques and technologies for effective humanitarian demining," *International Journal of Mine Action*, vol. 6, no. 1, 2002.
- [3] W. D. Jong and K. Schutte, "Processing of polarimetric infrared images for landmine detection," in *Proceedings of the 2nd International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, 2003.
- [4] B. N. Nelson, "Region of interest identification, feature extraction, and information fusion in a forward looking infrared sensor used in landmine detection," in *Proceedings of the IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications (CVBVS 2000)*. Washington, DC, USA: IEEE Computer Society, 2000, p. 94.
- [5] C. Bruschini, K. D. Bruyn, H. Sahli, and J. Cornelis, *EUDEM: The EU in Humanitarian DEMining*. EU report, Brussels, 1999.
- [6] S. Burke, "The u.s. department of defense humanitarian demining research and development program," *Journal of Mine Action*, vol. 7, no. 1, 2003.
- [7] C. P. Gooneratne, S. C. Mukhopahyay, and G. S. Gupta, "A review of sensing technologies for landmine detection:unmanned vehicle based approach," in *Proceedings of the 2nd International Conference on Autonomous Robots and Agents*, December 13-15, 2004 Palmerston North, New Zealand, 2004, pp. 401–407.
- [8] R. Harman, J. Holloway, and J. Broach, "Detection and remediation technologies for mines and minelike targets," in *Proceeding of the SPIE*, 2003.
- [9] L. Kempen, "Physical principals for anti-personal mine detection: A survey of three sensing principals," Technical Report IRIS-TR-0047, Department of Electronics and Information Processing, Vrije Universiteit Brussel, Tech. Rep., 1997.
- [10] M. Salameh, "Analysis and design of a landmine detection system using mobile devices," Master's thesis, Al-Balqa Applied University, 2008.
- [11] A. Ajlouni, "Improving landmine detection process using image processing techniques," Master's thesis, Al-Balqa Applied University, 2008.
- [12] A. Ajlouni and A. Sheta, "Landmind detection with IR sensors using karhunen loeve transformation and watershed segmentation," in *The 5th IEEE International Multi-Conference on Systems, Signals and Devices*, 2008, pp. 1–6.
- [13] T.-H. Chen, Y.-F. Lin, and T.-Y. Chen, "Intelligent vehicle counting method based on blob analysis in traffic surveillance," *Proceedings of The Second International Conference on Innovative Computing, Information and Control (ICICIC '07)*, pp. 238–238, 2007.
- [14] J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [15] P. Brigger and J. R. Casas, "Morphological operators for image and video compression," *IEEE Trans. on Image Processing*, vol. 5, pp. 881–897, 1996.
- [16] N. J. Leite and S. J. F. Guimarães, "Morphological residues and a general framework for image filtering and segmentation," *EURASIP J. Appl. Signal Process.*, vol. 2001, no. 1, pp. 219–229, 2001.
- [17] K. Mikula, T. Preusser, and M. Rumpf, "Morphological image sequence processing," *Comput. Vis. Sci.*, vol. 6, no. 4, pp. 197–209, 2004.
- [18] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.
- [19] D. Schonfeld, "Optimal structuring elements for the morphological pattern restoration of binary images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 589–601, 1994.
- [20] W. Messelink, K. Schutte, A. Vossepoel, F. Cremer, J. Schavemaker, and E. den Breejen, "Feature based detection of landmines in infrared images," in *Proceeding on the SPIE Conference, Detection and Remediation Technologies for Mines and Minelike Targets VII*, J. T. Broach, R. S. Harmon, and G. J. Dobeck, eds., (Orlando (FL), USA), vol. 4742, 2002, pp. 108–119.
- [21] J. Hervas and P. L. Rosin, "Image thresholding for landslide detection by genetic programming," in *Proceedings of the First International Workshop on Multitemporal Remote Sensing Images*, L. Bruzzone and P. Smits, Eds. University of Trento, Italy: World Scientific Publishing, 13-14 Sept. 2001.
- [22] P. sung Liao, T. sheng Chen, and P. choo Chung, "A fast algorithm for multilevel thresholding," *Journal of Information Science and Engineering*, vol. 17, pp. 713–727, 2001.
- [23] P. L. Rosin and E. Ioannidis, "Evaluation of global image thresholding for change detection," *Pattern Recogn. Lett.*, vol. 24, no. 14, pp. 2345–2356, 2003.
- [24] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, pp. 146–165, January 2004.
- [25] Y. Bazi, L. Bruzzone, and F. Melgani, "Image thresholding based on the EM algorithm and the generalized gaussian distribution," *Pattern Recogn.*, vol. 40, no. 2, pp. 619–634, 2007.
- [26] N. I. Corporation, "Image analysis and processing," <http://zone.ni.com/devzone/cda/tut/p/id/3470>, Sep 6, 2006, [Online; accessed 6-February-2008].
- [27] M. K. Habib, "Humanitarian demining: Reality and the challenge of technology-the state of the arts," *International Journal of advances Robotic systems*, vol. 4, no. 2, 2007.
- [28] P. Santana, J. Barata, O. Cruz, A. Mestre, J. Lisboa, and L. Flores, "A multi-robot system for landmine detection," in *Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2005)*, 2005, pp. 721–728.
- [29] H. Goldenberg, H. Najjaran, N. Kircanski, and A. A. Goldenberg, "Landmine detection using an autonomous terrain-scanning robot," *International Journal of Industrial Robot*, vol. 32, pp. 240–247, 2005.
- [30] Nokia, "Forum nokia website," <http://www.forum.nokia.com/>, 2007.
- [31] K. Willekens, "Meerdael (belgium) minefields images," http://apl-database.jrc.it/Home/Optical-IR/Non-JRC/RMA_Meerdael/, December 20 1999, [Online; accessed 6-February-2008].
- [32] M. Achery, D. Milojevic, Y. Yvinec, A. Pizurica, L. VanKempen, and N. Milisavljevic, "HUDEM IR trials at meerdael," <http://www.sic.rma.ac.be/nada/Meerdael.html>, April 1-4, 1998, [Online; accessed 6-February-2008].



Biography



Alaa F. Sheta received his B.E., M.Sc. degrees in Electronics and Communication Engineering from the Faculty of Engineering, Cairo University in 1988 and 1994, respectively. He received his Ph.D. degree from the Computer Science Department, School of Information Technology and Engineering, George Mason University, Fairfax, VA, USA in 1997. He was promoted to professorship in 2008. Currently, Prof. Sheta is a faculty member with the Computer Science Department, The World Islamic Sciences and Education (WISE) University, Amman, Jordan. He is on leave from the Computers and Systems Department, Electronics Research Institute (ERI), Cairo, Egypt. He published over 80 papers, book chapters and two books in the area of image processing and evolutionary computations. He has been an invited speaker in number of national and international conferences. Prof. Sheta is a member of the IEEE Evolutionary Computations, ACM and ISAI societies. He is also the Vice President of the Arab Computer Society (ACS). His research interests include Evolutionary Computation, Modeling and Simulation of Dynamical Nonlinear Systems, Image Processing, Robotics, Swarm Intelligence, Automatic Control, Fuzzy Logic, Neural Networks and Software Reliability Modeling.





Advanced D-Partitioning Stability Analysis of Digital Control Systems with Multivariable Parameters

Kamen M. Yanev¹

Department of Electrical Engineering, University of Botswana, Gaborone, Botswana

yanevkm@yahoo.com, yanevkm@mopipi.ub.bw

<http://www.ub.bw/>

Abstract

This paper contributes in further advancement of the method of D-Partitioning stability analysis applied to digital control system with multivariable parameters. It also explores the effects of simultaneous system uncertainties by determining graphically regions of stability in the space of the system's parameters. The Euler's approximation is taken into account and the Bilinear Tustin Transform method is employed as one of the most precise facilities for systems discretization. The interaction between the variable parameters will bring a new light in the graphical solution of the problem of control system stability, where the analysis of control systems with two and three simultaneously variable parameters is taken into consideration. The proposed research is a continuation and further development of the already published author's work on the advancement of the method of the D-Partitioning stability analysis of digital control systems with variable parameters. The suggested implementation for system's stability analysis is fundamental for the further advancement of control theory in this area.

Keywords: *Multivariable Parameters, Stability Analysis Tool, D-Partitioning, Interaction, Regions of Stability;*

Nomenclature

$G_O(s)$	Open-loop transfer function of the system
$G(s)$	Characteristic equation of the continuous system
s	Laplace operator
K	Variable gain of the control system
T_1, T_2, T_3	Variable time constants of the control system
T_s	Sampling period
T_{min}	Minimum time-constant of the system
ω_s	Sampling frequency
z	z-transform operator
μ, γ	System's variable parameters
2-D	Two-dimensional
3-D	Three-dimensional
GM	Gain margin
PM	Phase margin

1. Introduction

The D-partitioning method enables a quick and convenient determination of the regions of stability in case of variation of system's parameters. Following some initial ideas of Neimark [1], the method was better clarified and further developed by the author in previous published work [2], [3], [4], [5], [6], [7], now calcified as Advanced D-Partitioning. After its further expansion, the method became a powerful tool for system stability analysis, having a number of advantages compared to other well-known stability analysis methods. It has the benefit of a clear graphical display of the variation of each parameter and its effect on the system's stability.

The current research is extending the strategy for stability analysis of digital control systems with multivariable parameters, suggesting a technique for solving the problem by applying the Advanced D-Partitioning method in the discrete time-domain. The Advanced D-Partitioning analysis of digital control systems with multivariable parameters is achieved by introducing discretization based on the Bilinear Tustin Transform [8], while the systems' sampling time is chosen, considering the Euler's approximation [9], [10].

Exploring control systems with simultaneous variation of multiple parameters was inspired by cases of abnormal behaving systems, being stable at lower parameter values, further becoming unstable at a specific parameter range and turn out to be again stable again at higher parameter values [11], [12].

For the clarity of the presented research, this paper is organized in the following sequence. Initially, the nature of the D-Partitioning as stated by Neimark is explained. This is followed by further upgrading of the Advanced D-Partitioning analysis suggested by the author in his previous published work, now extended in the discrete time-domain. The results, achieved from the Advanced D-Partitioning analysis in the discrete-time domain for the cases of one, two and three variable parameters, are confirmed by the Nyquist and Bode stability criteria. The employed stability analysis methods are finally compared for the considered control systems variable parameters.



2. The Nature of the D-Partitioning as Originally Suggested by Neimark

The position of the characteristic equation roots in the s-plane depends on the values of the system's parameters. On this basis, Neimark suggested that the space of an n -order system's characteristic equation coefficients can be partitioned into a number of regions corresponding to the number of roots in the left-hand side of the s-plane. This method was categorized as D-Partitioning. At its initial steps the method was stating rather its theoretical possibilities and its applications were quite limited. Although, Neimark expanded his initial ideas, they were rarely implemented because of their obscurity.

3. Advanced D-Partitioning Stability Analysis (Case of One Variable Parameter)

As already indicated, in previous published work, the author of this research suggested further advancement of the method of the D-Partitioning stability analysis of continuous control systems by one variable parameter. To facilitate and realize the stability analysis, the system's characteristic equation is presented in the format [2], [3]:

$$G(s) = a_0 s^n + a_1 s^{n-1} + \dots + a_n = 0 \quad (1)$$

Equation (1) can be also introduced in the following format to expose the variable parameter:

$$G(s) = P(s) + vQ(s) = 0 \quad (2)$$

The D-partitioning regions could be obtained by substituting $s = j\omega$.

$$G(j\omega) = P(j\omega) + vQ(j\omega) = 0 \quad (3)$$

Therefore, the variable parameter can be presented as a complex number:

$$v = -\frac{P(j\omega)}{Q(j\omega)} = X(\omega) + jY(\omega) \quad (4)$$

The D-partitioning regions are obtained graphically in the v-plane by allocating values of the frequency within the range $-\infty \leq \omega \leq +\infty$.

Since, in the s-plane, the region of stability remains on the left-hand side of the plane, in the complex plane $v = X(\omega) + jY(\omega)$, a region of stability remains always on the left-hand side of the D-Partitioning curve for a change of frequency from $-\infty$ to $+\infty$.

The D-partitioning curve in terms of one variable parameter can be plotted in the complex plane within the frequency range $-\infty \leq \omega \leq +\infty$, facilitated by MATLAB the "nyquist" m-code. The procedure can work on any computer where the MATLAB program is installed.

To avoid any misunderstanding or misinterpretation of the D-Partitioning procedure, the "nyquist" m-code is modified into a "dpartition" m-code with the aid of the MATLAB Editor and a proper formatting. The "dpartition" m-code will plot the curve of a specific system parameter in terms of the frequency variation from $-\infty$ to $+\infty$.

The results obtained by the "dpartition" m-code are the same as those achieved by the "nyquist" m-code. This D-Partitioning procedure can work only on a computer where the new developed "dpartition" m-code is included in the MATLAB program.

In order to benefit from the Advanced D-Partitioning analysis shown in this research, the wider engineering community can still use the "nyquist" m-code for the purpose of plotting the D-partitioning curve.

The Advanced D-Partitioning analysis in case of one variable parameter is illustrated in this research by a system consisting of an armature-controlled dc motor and a type-driving mechanism is suggested [11], [12], [13]. Initially, it is considered that the system is experiencing variation of only one of its parameters. Usually, the variation of the load causes variation of the mechanism time-constant T_l . The transfer function of the open loop system in the continuous-time domain is presented as:

$$G_{O1}(s) = \frac{10}{(1 + T_1 s)(1 + 0.5s)(1 + 0.8s)} = \frac{10}{T_1 (0.4s^3 + 1.3s^2 + s) + 0.4s^2 + 1.3s + 1} \quad (5)$$

The characteristic equation of the continuous stand-alone unity feedback system is:

$$G(s) = T_1 (0.4s^3 + 1.3s^2 + s) + 0.4s^2 + 1.3s + 11 = 0 \quad (6)$$

The regions of stability reflecting the variation of the system's parameter T_l are determined by implementing the method of the Advanced D-Partitioning. The time-constant T_l is presented from equation (6) as follows:

$$T_1(s) = -\frac{T_1 T_2 s^2 + (T_1 + T_2)s + K + 1}{T_1 T_2 s^3 + (T_1 + T_2)s^2 + s} = -\frac{0.4s^2 + 1.3s + 11}{0.4s^3 + 1.3s^2 + s} \quad (6)$$

The variable $T_l(s)$ is initially introduced as a continuous-time function and next converted into its digital equivalent $T_l(z)$, facilitated by the Tustin Transform [14], [15], [16]. In accordance with the Euler's approximation, observing that $T_s \leq (0.1T_{min} \text{ to } 0.2T_{min})$, the sampling period is chosen as $T_s = 0.05\text{sec}$, since the continuous system minimum time-constant is $T_{min} = 0.5\text{sec}$. Then, the D-Partitioning in terms of the variable time-constant $T_l(z)$ is achieved in the discrete-time domain by the code:

```
>> T1 = tf([-0.4 -1.3 -11],[0.4 1.3 1 0])
Transfer function:
-0.4 s^2 - 1.3 s - 11
-----
0.4 s^3 + 1.3 s^2 + s
>> T1d = c2d(T1,0.05,'tustin')
Transfer function:
-0.02536 z^3 + 0.02002 z^2 + 0.02377 z - 0.02161
-----
z^3 - 2.844 z^2 + 2.694 z - 0.8499
Sampling time: 0.05
>> dpartition(T1d)
```



From where:

$$T_1(z) = -\frac{-0.02536z^3 + 0.02002z^2 + 0.02377z - 0.02161}{z^3 - 2.844z^2 + 2.694z - 0.8499} \quad (7)$$

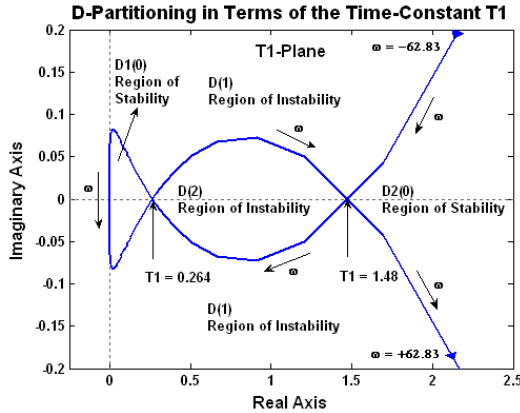


Figure 1. Advanced D-Partitioning Analysis of the Digital Control System in Terms of the Variable Time-Constant T_1

As seen from Figure 1, the Advanced D-Partitioning determines four regions in the complex T_1 -Plane: D1(0), D2(0), D(1) and D(2). Only D1(0) and D2(0) are regions of stability, since the position of these regions is always on the left-hand side of the D-Partitioning curve within the range of $\omega = \pm \omega_s/2 = \pm 2\pi/2T_s = \pm 62.83$ rad/sec [17]. Only the real values of T_1 are considered, hence the regions of stability are reduced to lines of stability.

4. System Evaluation with Nyquist and Bode Stability Criteria (Case of One Variable)

If the time-constant is within $0 \text{ sec} \leq T_1 \leq 0.264 \text{ sec}$, corresponding to the region of stability D1(0), the system will be stable. If T_1 is varied within $0.264 \text{ sec} \leq T_1 \leq 1.48 \text{ sec}$, the system will be unstable, being within the region of instability D(2). If $T_1 > 1.48 \text{ sec}$, the system is again stable and is operating in the region of stability D2(0). If the time-constant is $T_1 = 0.264 \text{ sec}$ or $T_1 = 1.48 \text{ sec}$, the digital closed-loop control system becomes marginal.

The open-loop transfer functions for both marginal cases of the control system in the continuous time-domain are determined accordingly:

$$G_{O.0.264}(s) = G_{O1}(s) = \frac{10}{(1 + 0.264s)(1 + 0.5s)(1 + 0.8s)} = \frac{10}{0.1056s^3 + 0.7432s^2 + 1.564s + 1} \quad (8)$$

$$G_{O.1.48}(s) = G_{O1}(s) = \frac{10}{(1 + 1.48s)(1 + 0.5s)(1 + 0.8s)} = \frac{10}{0.592s^3 + 2.324s^2 + 2.78s + 1} \quad (9)$$

The Advanced D-Partitioning in the discrete-time domain is confirmed with the Nyquist stability criterion and the Bode stability criterion for the two marginal values of T_1 . They are introduced into the system's continuous transfer functions that are further converted into their digital equivalents [8], [13], [14], [17] with the Bilinear Tustin Transform. This is facilitated by the following code:

```
>> Go1=tf([0 10],[0.1056 0.7432 1.564 1])
Transfer function:
10

0.1056 s^3 + 0.7432 s^2 + 1.564 s + 1
>> God1 = c2d(Go1,0.05,'tustin')
Transfer function:
0.001248 z^3 + 0.003745 z^2 + 0.003745 z + 0.001248

z^3 - 2.671 z^2 + 2.375 z - 0.7029
Sampling time: 0.05
>> Go2=tf([0 10],[0.592 2.324 2.78 1])
Transfer function:
10

0.592 s^3 + 2.324 s^2 + 2.78 s + 1
>> God2 = c2d(Go2,0.05,'tustin')
Transfer function:
0.0002397 z^3 + 0.0007191 z^2 + 0.0007191 z + 0.0002397

z^3 - 2.811 z^2 + 2.633 z - 0.8217
Sampling time: 0.05
>> nyquist(God1,God2)
>> bode(God1,God2)
>> margin(Gpod1)
GM = 0.00652 dB, PM = 0.0193 deg
>> margin(Gpod2)
GM = -0.0756 dB, PM = -0.276 deg
```

As seen, after applying the Bilinear Tustin Transform, the digital equivalents of equations (8) and (9) are reflected in equations (10) and (11), as seen from the code above.

$$G_{Od.0.264}(z) = G_{od1}(z) = \frac{0.001248z^3 + 0.003745z^2 + 0.003745z + 0.001248}{z^3 - 2.671z^2 + 2.375z - 0.7029} \quad (10)$$

$$G_{Od.1.48}(z) = G_{od2}(z) = \frac{0.0002397z^3 + 0.0007191z^2 + 0.0007191z + 0.0002397}{z^3 - 2.811z^2 + 2.633z - 0.8217} \quad (11)$$

As a result of the applied code, Figure 2 demonstrates the Nyquist stability criterion in the discrete-time domain for both of the cases related to marginal time-constant values $T_1 = 0.264 \text{ sec}$ and $T_1 = 1.48 \text{ sec}$.

Marginal Nyquist Diagrams in the Discrete Time-Domain
Cases of $T_1 = 0.264 \text{ sec}$ and $T_1 = 1.48 \text{ sec}$

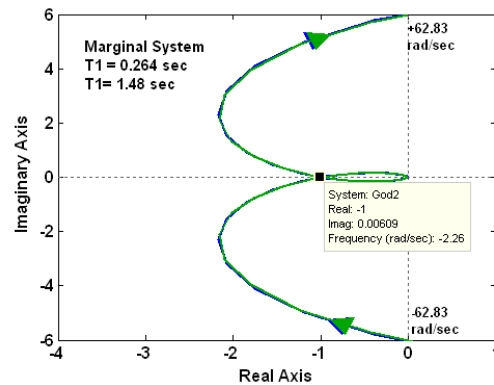


Figure 2. Confirmation of the Result of the Advanced D-Partitioning Analysis with the Aid of the Nyquist Stability Criterion in the Discrete-Time Domain (Marginal Cases $T_1 = 0.264 \text{ sec}$; $T_1 = 1.48 \text{ sec}$)



As seen from Figure 2, the Nyquist curves of the digital open loop system coincide and are passing approximately via the point $(-1, j0)$, hence the closed loop digital control system is marginal. This confirms the results, obtained from the Advanced D-Partitioning stability analysis.

The achieved results from the code, confirm also the outcome of the Advanced D-Partitioning analysis with the aid of the Bode stability criterion in the discrete-time domain for both of the cases related to marginal time-constant values $T_1 = 0.264$ sec and $T_1 = 1.48$ sec.

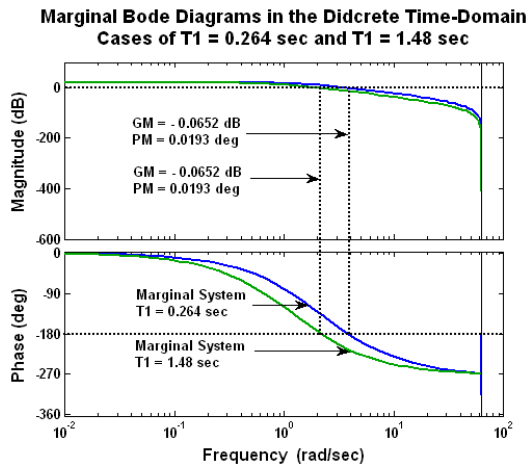


Figure 3. Confirmation of the Result of the Advanced D-Partitioning Analysis with the Aid of the **Bode Stability Criterion** in the Discrete-Time Domain (Marginal Cases $T_1 = 0.264$ sec; $T_1 = 1.48$ sec)

As shown in Figure 3, if the marginal value of the time-constant is $T_1 = 0.264$ sec, the results for the digital system gain margin is $GM = 0.00652$ dB ≈ 0 dB and the system phase margin is $PM = 0.0193^\circ \approx 0^\circ$. The results also prove the second marginal value of the time-constant $T_1 = 1.48$ sec, where the gain margin is $GM = -0.0756$ dB ≈ 0 dB and phase margin is $PM = -0.276^\circ \approx 0^\circ$.

Again, the analysis of both Nyquist stability criterion and Bode stability criterion is restricted to the frequency $\omega = \pm \omega_s/2 = \pm 2\pi/2T_s = \pm 62.83$ rad/sec [11], [12], [17].

The system stability within the regions D1(0) and D2(0) is explored in the discrete-time domain by allocating values for the variable time-constant $T_1 = 0.1$ sec and $T_1 = 5$ sec and applying the following code:

```
>> Go3=tf([0 10],[0.04 0.53 1.4 1])
>> God3=c2d(Go3,0.05,'tustin')
Transfer function:
0.002886 z^3 + 0.008658 z^2 + 0.008658 z + 0.002886
-----
z^3 - 2.444 z^2 + 1.956 z - 0.51
Sampling time: 0.05
>> Go4=tf([0 10],[2 6.9 6.3 1])
>> God4=c2d(Go4,0.05,'tustin')
Transfer function:
7.179e-005 z^3 + 0.0002154 z^2 + 0.0002154 z + 7.179e-005
-----
z^3 - 2.834 z^2 + 2.676 z - 0.8415
Sampling time: 0.05
>> nyquist(God3,God4)
>> bode(God3,God4)
>> margin(God3)
GM = 4.89 dB, PM = 15.5 deg
>> margin(God4)
GM = 6.34 dB, PM = 24 deg
```

**Nyquist Diagrams in the Discrete Time-Domain
Cases of Stable System; $T_1 = 0.1$ sec and $T_1 = 5$ sec**

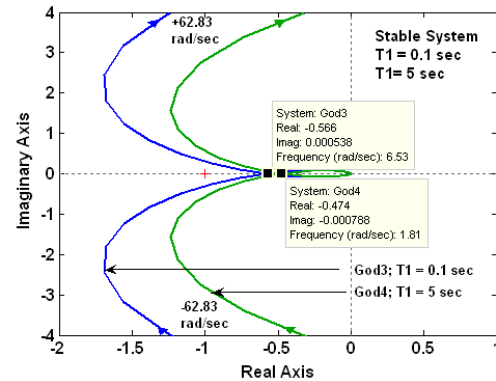


Figure 4. Confirmation of the Result of the Advanced D-Partitioning Analysis with the Aid of the **Nyquist Stability Criterion** in the Discrete-Time Domain (Stable Case $T_1 = 0.1$ sec; $T_1 = 5$ sec)

Since, the Nyquist curves of the digital open loop control system are passing approximately via the points at the real axes $(-0.566, j0)$ and $(-0.474, j0)$, the closed loop digital control system is stable for both cases, confirming the result from the Advanced D-Partitioning analysis.

Bode Diagram in the Discrete Time-Domain (Case $T_1 = 0.1$ sec)
Gm = 4.89 dB (at 5.87 rad/sec), Pm = 15.5 deg (at 4.46 rad/sec)

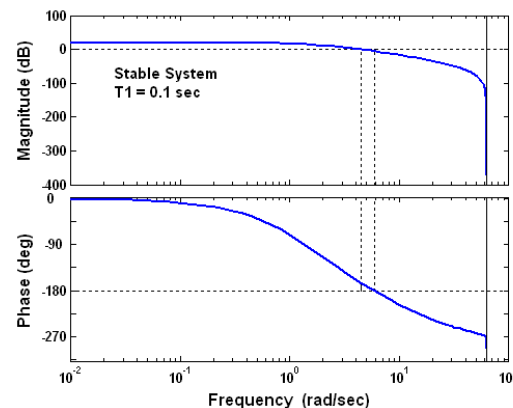


Figure 5. Confirmation of the Result of the Advanced D-Partitioning Analysis with the Aid of the **Bode Stability Criterion** in the Discrete-Time domain (Case of Stable System $T_1 = 0.1$ sec)

Bode Diagram in the Discrete-Time (Case $T_1 = 5$ sec)
Gm = 6.34 dB (at 1.77 rad/sec), Pm = 24 deg (at 1.21 rad/sec)

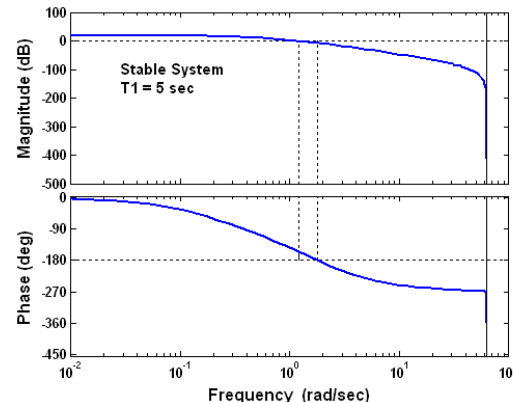


Figure 6. Confirmation of the Result of the Advanced D-Partitioning Analysis with the Aid of the **Bode Stability Criterion** in the Discrete-Time domain (Case of Stable System $T_1 = 5$ sec)



As seen from Figure 5 and Figure 6, and from the results of the code, the Advanced D-Partitioning analysis is confirmed with the aid of the Bode stability criterion in the discrete-time domain for both of the cases related to a stable system ($T_1 = 0.1\text{sec}$ and $T_1 = 5\text{sec}$). The positive gain and phase margins ($GM = 4.89\text{ dB}$ and $PM = 15.5^\circ$) and ($GM = 6.34\text{ dB}$ and $PM = 24^\circ$), prove that the digital system is stable within the regions D1(0) and D2(0).

Further, the system stability within the region D(2) is explored in the discrete-time domain by allocating a value for the variable $T_1 = 1\text{ sec}$ and applying the code:

```
>> Go5=tf([0 10],[0.4 1.7 2.3 1])
>> God5 = c2d(Go5,0.05,'ustin')
Transfer function:
0.000352 z^3 + 0.001056 z^2 + 0.001056 z + 0.000352
-----
z^3 - 2.795 z^2 + 2.604 z - 0.8085
Sampling time: 0.05
>> nyquist(God5)
>> bode(God5)
>> margin(God5)
GM = -1.14 dB, PM = -4.11deg
```

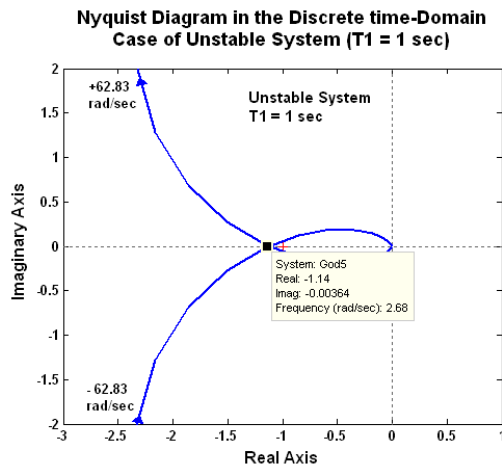


Figure 7. Confirmation of the Result of the Advanced D-Partitioning Analysis with the Aid of the **Nyquist Stability Criterion** in the Discrete-Time domain (Unstable Case $T_1 = 1\text{ sec}$)

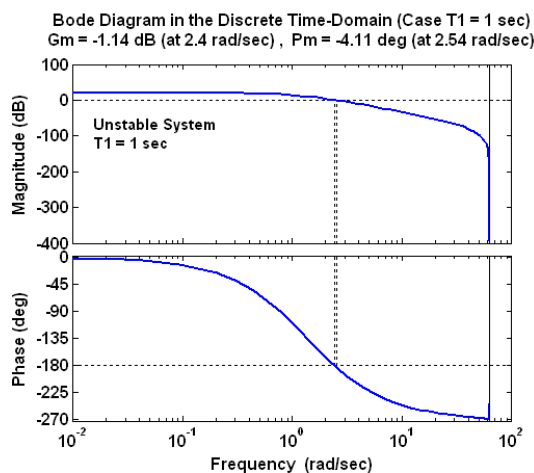


Figure 8. Confirmation of the Result of the Advanced D-Partitioning Analysis with the Aid of the **Bode Stability Criterion** in the Discrete-Time domain (Unstable Case $T_1 = 1\text{ sec}$)

If applying the Nyquist stability criterion for the case $T_1 = 1$, the Nyquist curve of the digital open loop control system is passing approximately via the points at the real axes $(-1.14, j0)$. Therefore, the closed loop digital control system is unstable.

By applying the Bode stability criterion for the case $T_1 = 1$, the negative values of the digital system margins ($GM = -1.14\text{ dB}$ and $PM = -4.11^\circ$), confirm that the digital system is unstable within the region D(2).

In case of control systems with one variable parameter, the Advanced D-Partitioning method directly exposes the transparent images of regions of stability and instability, as well as the system's margins of stability in the complex plane of the variable parameter. It is also demonstrated that the Advanced D-Partitioning can be successfully applied to digital control systems. Both Nyquist and Bode stability criteria confirm the results from the Advanced D-Partitioning analysis.

5. Advanced D-Partitioning Stability Analysis (Case of Two Variable Parameters)

Further upgrade of the Advanced D-Partitioning analysis in case of systems with two simultaneously variable parameters [5], [6], [7] is suggested in this research. The system's characteristic equation generally is presented as:

$$G(s) = \mu P(s) + \gamma Q(s) + R(s) = 0 \quad (12)$$

where $P(s)$, $Q(s)$, and $R(s)$ are polynomials of s and μ and γ are system's variables parameters

The border of the D-Partitioning regions in the plain (μ, γ) is determined by:

$$G(j\omega) = \mu P(j\omega) + \gamma Q(j\omega) + R(j\omega) = 0 \quad (13)$$

If a hypothetical third order unity feedback system of Type 0 is considered, the system's characteristic equation can be presented as:

$$G(s) = (T_1 s + 1)(T_2 s + 1)(T_3 s + 1) + K = 0 \quad (14)$$

It is suggested that two of the system's parameters are simultaneously variable:

$$T_1 = T = \mu, \quad K = \gamma. \quad (15)$$

The objective is to determine the regions of variation of these two parameters, for which the system will be stable. Equations (15) are substituted in (14), from where:

$$\left. \begin{aligned} \mu [T_2 T_3 s^3 + (T_2 + T_3) s^2 + s] + \\ + \gamma + T_2 T_3 s^2 + (T_2 + T_3) s + 1 = 0 \end{aligned} \right\} \quad (16)$$

By substituting $s = j\omega$ in equation (16) and taking into account equation (13), equation (16) could be presented in the detailed form as set of equations (17):

$$\left. \begin{aligned} P(j\omega) &= [T_2 T_3 (j\omega)^3 + (T_2 + T_3) (j\omega)^2 + j\omega] \\ Q(j\omega) &= 1 \\ R(j\omega) &= T_2 T_3 (j\omega)^2 + (T_2 + T_3) j\omega + 1 \end{aligned} \right\} \quad (17)$$

Taking into consideration equations (13) and (17), the variable system's parameters can be determined as:



$$\left. \begin{aligned} \mu &= \frac{\Delta_1}{\Delta} = \frac{T_2 + T_3}{T_2 T_3 \omega^2 - 1} \\ \gamma &= \frac{\Delta_2}{\Delta} = \frac{(T_2 T_3 \omega^2 - 1)^2 + (T_2 + T_3)^2 \omega^2 + 1}{T_2 T_3 \omega^2 - 1} \end{aligned} \right\} \quad (18)$$

$$\text{or } \Delta = T_2 T_3 \omega^2 - 1 \quad (19)$$

The determinant Δ becomes $\Delta = 0$ at a specific frequency $\omega = \omega_\infty$ that can be found out from equation (19) as:

$$\omega = \omega_\infty = \sqrt{\frac{1}{T_2 T_3}} \quad (20)$$

If $\Delta = 0$, it is clear from equations (20) that at $\omega = \omega_\infty$ both system parameters are approaching infinity:

$$\mu(\omega_\infty) \rightarrow \infty, \quad \gamma(\omega_\infty) \rightarrow \infty, \quad (21)$$

This implies that the main D-Partitioning curve has an interruption, or a breakdown, at a frequency $\omega = \omega_\infty$. It consists of two parts, plotted within the frequency ranges $0 < \omega < \omega_\infty$ and $\omega_\infty < \omega < \infty$.

For a better clarification and simplicity, first the functions $\mu(\omega)$ and $\gamma(\omega)$ are plotted, as shown in Figure 9(a) and Figure 9(b).

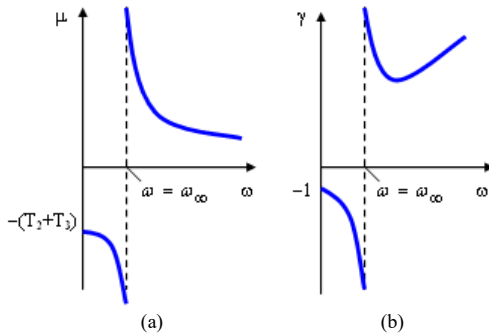


Figure 9. The Graphical Presentations of $\mu(\omega)$ and $\gamma(\omega)$ Showing the Interruption of the Curves at a Frequency $\omega = \omega_\infty$

The regions of the D-Partitioning also depend on straight lines in the (μ, γ) plane, known as *special lines*. The special lines are plotted for two border frequencies $\omega = 0$ and $\omega = +\infty$. Then the coefficients a_n and a_o of the equation (1) depend directly on the parameters μ and γ and the equations of the special lines are obtained by:

$$\left. \begin{aligned} a_n &= \mu T_2 T_3 = 0, & a_o &= \gamma + 1 = 0 \\ \mu &= 0 = \text{Const} & \gamma &= -1 = \text{Const} \end{aligned} \right\} \quad (22)$$

By combining the curves $\mu(\omega)$, $\gamma(\omega)$ and the special lines, the D-Partitioning is obtained in the (μ, γ) plane as seen in Figure 10. Taking into account that $\mu = T_1 = T$ is a time-constant and it can adopt only positive values, only the stable region $D_1(0)$ should be considered. The region of stability $D_1(0)$ is locked within the left-hand side of the D-Partitioning curve, corresponding to frequency rise from $\omega = \omega_\infty$ to $\omega \rightarrow \infty$ and the special line $\gamma = -1$. Since the gain $K = \gamma$ may also adopt only positive values, the realistic border of the stable region $D_1(0)$ should be considered $K = \gamma = 0$.

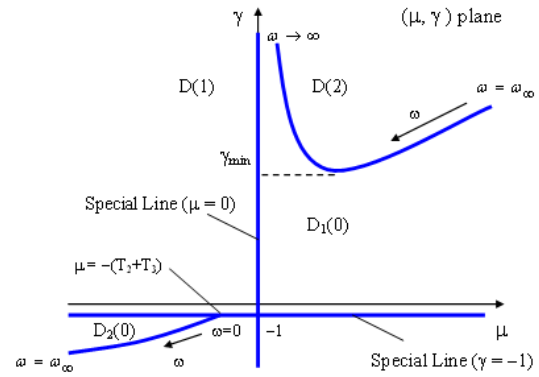


Figure 10. The Regions of Stability $D_1(0)$ and $D_2(0)$ Locked between the D-Partitioning Curves and the Special Lines

If the gain $K < \gamma_{min}$, the system is stable for any values of the time-constant $\mu = T_1$. For $K > \gamma_{min}$, the system is stable only for very small or very large values of the time-constant $\mu = T_1$. This case is demonstrated for the system of the armature-controlled dc motor and a type-driving mechanism with open-loop transfer function:

$$G_{O2}(s) = \frac{K}{(1 + Ts)(1 + 0.5s)(1 + 0.8s)} \quad (23)$$

The characteristic equation of the feedback system is:

$$K + (1 + Ts)(1 + 0.5s)(1 + 0.8s) = 0 \quad (24)$$

By substituting $s = j\omega$ equation (24) is modified to:

$$K = -1 + (1.3T + 0.4)\omega^2 + j\omega(0.4T\omega^2 - 1.3 - T) \quad (25)$$

Since the gain may obtain only real values, the imaginary term of equation (25) is set to zero, from where:

$$\omega^2 = \frac{1.3 + T}{0.4T} \quad (26)$$

The result of (26) is substituted into the real part of equation (25), from where:

$$K = \frac{1.3T^2 + 1.69T + 0.52}{0.4T} = 3.25T + 4.225 + \frac{1.3}{T} \quad (27)$$

When considering the continuous control system with the transfer function of equation (23), the D-Partitioning curve $K = f(T)$ is plotted with the aid of the code:

```
>> T = 0:0.1:5;
>> K = 3.25.*T+4.225+1.3./T
K =
Columns 1 through 10
Inf 17.5500 11.3750 9.5333 8.7750 8.4500 8.3417
8.3571 8.4500 8.5944
Columns 11 through 20
8.7750 8.9818 9.2083 9.4500 9.7036 9.9667 10.2375
10.5147 10.7972 11.0842
Columns 21 through 30
11.3750 11.6690 11.9659 12.2652 12.5667 12.8700
13.1750 13.4815 13.7893 14.0983
Columns 31 through 40
14.4083 14.7194 15.0313 15.3439 15.6574 15.9714
16.2861 16.6014 16.9171 17.2333
Columns 41 through 50
17.5500 17.8671 18.1845 18.5023 18.8205 19.1389
19.4576 19.7766 20.0958 20.4153
Column 51
20.7350
>> plot(T,K)
```



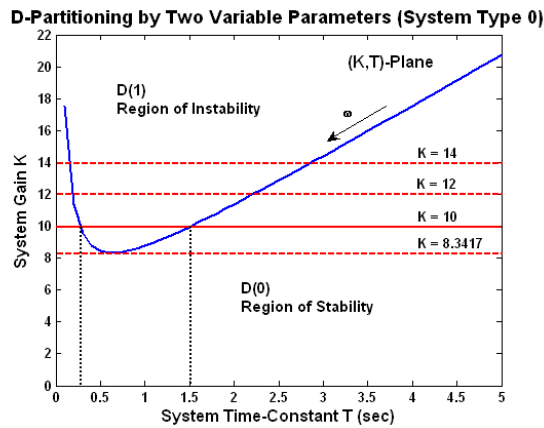


Figure 11. D-Partitioning in Terms of Two Variable Parameters

As seen from Figure 11, if $K < 8.3417$, a limit determined with the aid of MATLAB interactive procedure, the system is stable for any value of the T . The system performance and stability depends on the interaction between the two simultaneously varying parameters.

Further, a very important phenomenon is demonstrated in Figure 11. It is seen that the D-Partitioning curve $K = f(T)$ defines the border between the region of stability $D(0)$ and instability $D(1)$ for the case of simultaneous variation of the two system parameters. **Each point of the D-Partitioning curve represents the simultaneous marginal values of the two variable parameters. This result is a unique advancement and innovation in the theory of control systems stability analysis.**

The system performance is investigated for any values of the variable gain K , like $K = 10$, $K = 12$, $K = 14$. If K is varied, this affects the values of T at which the system may become unstable. Higher values of the gain K , enlarges the range of the time-constant T at which the system will fall into instability.

The parameters interaction of the continuous control system is demonstrated in case of variation of the time-constant T , when the gain is set to $K = 10$. If the time-constant range is $0 < T < 0.25$ sec, or $T > 1.5$ sec the system is stable. The system becomes unstable in the time-constant range $0.25 \text{ sec} < T < 1.5$ sec.

If the results of the continuous control system are compared with the results for the same system in the digital time-domain, as demonstrated in Figure 1, it is obvious that there is a very close match. In the digital time-domain within the range of $\omega = \pm \omega/2 = \pm 2\pi/2T_s$, if the gain is set to $K = 10$, the system is stable for the time intervals $0 < T < 0.264$ sec, or $T > 1.48$ sec. It is unstable in the range $0.264 \text{ sec} < T < 1.48$ sec. This insignificant difference of the results is due to the implementation of the Euler's approximation.

6. System Evaluation with Nyquist and Bode Stability Criteria (Case of Two Variables)

The performance of the control system with two variable parameters can be tested with the Nyquist stability criterion and the Bode stability criterion in the discrete-time domain for any of the simultaneous marginal values

of the variables T and K . Initially, the system's transfer function is introduced as a continuous-time function and next it is converted into its digital equivalent.

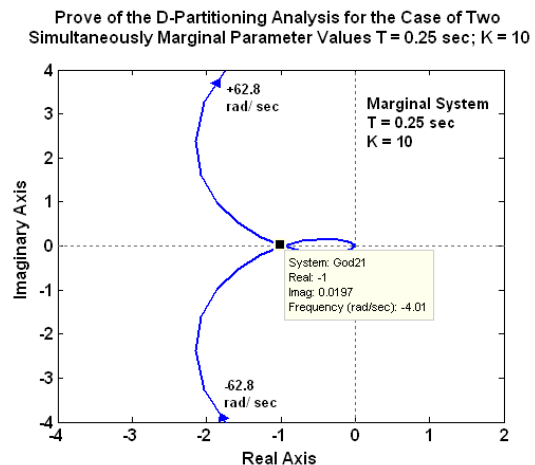
By substituting any two simultaneous marginal values of the two variable parameters, for instance, time-constant $T = 0.25$ sec and gain $K = 10$ in equation (23) results in:

$$G_o(s) = \frac{10}{(1+0.25s)(1+0.5s)(1+0.8s)} = \frac{10}{0.1s^3 + 0.725s^2 + 1.55s + 1} \quad (28)$$

Again, taking into account the Euler's approximation, $T_s \leq (0.1T_{min} \text{ to } 0.2T_{min})$, where the system's minimum time-constant is $T_{min} = 0.5$ sec and the sampling period is chosen as $T_s = 0.05$ sec. The system's stability assessment in the discrete-time domain is achieved by the code:

```
>> Go21=tf([0 10],[0.1 0.725 1.55 1])
Transfer function:
      10
-----
0.1 s^3 + 0.725 s^2 + 1.55 s + 1
>> God21 = c2d(Go21,0.05,'tustin')
Transfer function:
0.001312 z^3 + 0.003935 z^2 + 0.003935 z + 0.001312
-----
z^3 - 2.662 z^2 + 2.359 z - 0.6954
Sampling time: 0.05
>> nyquist(God21)
>> margin(God21)
GM = 0.017 dB, PM = 0.028 deg
```

When applying the Nyquist stability criterion, as seen from Figure 12, the Nyquist curve of the digital open loop system is passing approximately via the point $(-1, j0)$, therefore the closed loop digital control system is marginal. This confirms the results, obtained from the Advanced D-Partitioning stability analysis for the marginal system with two variable parameters.

Figure 12. Confirmation of the result of the Advanced D-Partitioning Analysis with the Aid of the Nyquist Stability Criterion in the Discrete-Time domain (Marginal Case $T = 0.25$ sec; $K = 10$)

As seen from the Bode diagram presented at Figure 13, the system is explored in the discrete-time domain for $T = 0.25$ sec and $K = 10$. The achieved results for the gain and phase margins are $GM = 0.204 \text{ dB} \approx 0 \text{ dB}$ and $PM = 0.729^\circ \approx 0^\circ$, proving that the system is marginal.



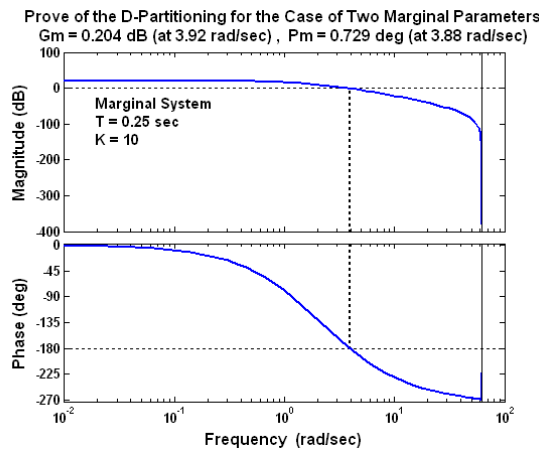


Figure 13. Prove of the D-Partitioning Analysis with the aid of the Bode Stability Criterion – Case of Two Simultaneously Marginal Values (Case $T = 0.25\text{sec}$ and $K = 10$)

The system stability within the region $D(0)$ is explored in the discrete-time domain if $T = 1$ sec and $K = 5$. The positive values for the gain and phase margins ($GM = 4.89$ dB and $PM = 20.6^\circ$), prove that the digital system is stable within the region $D(0)$, as seen from the code:

```
>> Go22=tf([0 5],[0.4 1.7 2.3 1])
Transfer function:
      5
-----
0.4 s^3 + 1.7 s^2 + 2.3 s + 1
>> God22 = c2d(Go22,0.05,'tustin')
Transfer function:
0.000176 z^3 + 0.0005279 z^2 + 0.0005279 z + 0.000176
-----
z^3 - 2.795 z^2 + 2.604 z - 0.8085
Sampling time: 0.05
>> margin(God22)
GM = 4.89 dB, PM = 20.6 deg
```

The system instability within the region $D(1)$ is explored in the discrete-time if $T = 1$ sec and $K = 14$. The negative values of the digital system margins ($GM = -4.06$ dB and $PM = -13.7^\circ$), prove that the digital system is unstable within the region $D(1)$, as seen from the code:

```
>> Go23=tf([0 14],[0.4 1.7 2.3 1])
Transfer function:
      14
-----
0.4 s^3 + 1.7 s^2 + 2.3 s + 1
>> God23 = c2d(Go23,0.05,'tustin')
Transfer function:
0.0004927 z^3 + 0.001478 z^2 + 0.001478 z + 0.0004927
-----
z^3 - 2.795 z^2 + 2.604 z - 0.8085
Sampling time: 0.05
>> margin(God23)
GM = -4.06 dB, PM = -13.7 deg
```

7. Advanced D-Partitioning Stability Analysis (Case of Three Variable Parameters)

Further advancement of the D-Partitioning is suggested in case of systems with three simultaneously variable parameters, where their interaction is presented in the 3-Dimensional (3-D) space. This is demonstrated for the control system of the armature-controlled dc motor and a type-driving mechanism, where the system's gain K and

two of the time-constants T_1 and T_2 are variable. The open-loop transfer function of the system is presented as:

$$G(s) = \frac{K}{(1 + T_1 s)(1 + T_2 s)(1 + 0.8s)} \quad (29)$$

The characteristic equation of the feedback system is:

$$K + (1 + T_1 s)(1 + T_2 s)(1 + 0.8s) = 0 \quad (30)$$

By substituting $s = j\omega$ equation (30) is modified to:

$$K = -1 + 0.8(T_1 + T_2)\omega^2 + j\omega(-T_1 - T_2 - 0.8 + 0.8T_1T_2\omega^2) \quad (31)$$

The imaginary term of equation (31) is set to zero, since the gain may obtain only real values, from where:

$$\omega^2 = \frac{T_1 + T_2 + 0.8}{0.8T_1T_2} \quad (32)$$

When the result of (32) is substituted into the real part of equation (31), results in:

$$K = -1 + 0.8(T_1 + T_2) + \frac{T_1 + T_2 + 0.8}{0.8T_1T_2} \quad (34)$$

The D-Partitioning plane is created in the (K, T_1, T_2) -Space of the variable parameters. The code is only partly presented, due to its considerable length:

```
>> [X,Y] = meshgrid([0:15:3])
X =
Columns 1 through 10
    0    0.1500    0.3000    0.4500    0.6000    0.7500    0.9000    1.0500
1.2000    1.3500
    0    0.1500    0.3000    0.4500    0.6000    0.7500    0.9000    1.0500
1.2000    1.3500
.....
Columns 11 through 20
1.5000    1.6500    1.8000    1.9500    2.1000    2.2500    2.4000
2.5500    2.7000    2.8500
1.5000    1.6500    1.8000    1.9500    2.1000    2.2500    2.4000
2.5500    2.7000    2.8500
Y =
Columns 1 through 10
    0    0    0    0    0    0    0    0    0    0
0.1500    0.1500    0.1500    0.1500    0.1500    0.1500    0.1500    0.1500
0.1500    0.1500    0.1500
0.3000    0.3000    0.3000    0.3000    0.3000    0.3000    0.3000    0.3000
0.3000    0.3000    0.3000
.....
Column 21
    0
0.1500
0.3000
>> Z = -1+0.8.*(X+Y)+(X+Y+0.8)./(0.8.*X.*Y)
Warning: Divide by zero.
Z =
Columns 1 through 10
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    60.3511    34.0822    25.4059    21.1278    18.6089    16.9696
15.8330    15.0106    14.3975
    Inf    34.0822    18.9244    13.9519    11.5256    10.1178    9.2193
8.6117    8.1861    7.8817
3.6249    3.5605

Column 21
    Inf
12.4922
7.3344
.....
4.7444
>> surf(X,Y,Z,gradient(Z))
```



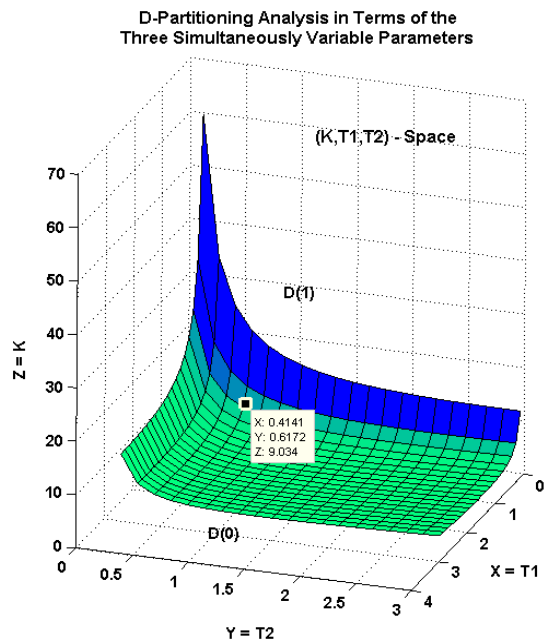


Figure 14. D-Partitioning in Case of a System with Three Simultaneously Variable Parameters

The D-Partitioning plane, as seen in Figure 14, is defining the border between the space of stability D(0), below the plane, and the space of instability D(1), above the plane, for the case of simultaneous variation of three system's parameters.

According to the Advanced D-Partitioning rule, in this case the space of stability is the one that remains always on the left-hand side of the D-Partitioning plane within the range of $\omega = \pm \omega/2 = \pm 2\pi/2T_s$. Following this rule, the stable space is proved to be the space D(0).

Any point of the D-partitioning plane represents the simultaneous marginal values of the three variable parameters, which is a unique advancement and in the theory of control systems stability analysis.

8. System Evaluation with Nyquist and Bode Stability Criteria (Case of Three Variables)

For the chosen point on the D-partitioning plane, shown in Figure 14, the system marginal gain is $K = Z = 9.034$ and the marginal values of the two time-constants are $T_1 = X = 0.4141$ sec and $T_2 = Y = 0.6172$ sec accordingly. To test the system's performance, its transfer function is introduced initially in the continuous-time domain and then converted into its digital equivalent. Substituting the three marginal values of the variable parameters in equation (29) results in:

$$G_{031}(s) = \frac{9.034}{(1 + 0.4141s)(1 + 0.6172s)(1 + 0.8s)} \approx \frac{9.034}{0.192s^3 + 1.04s^2 + 1.8s + 1} \quad (33)$$

The system stability within the space D(0) is explored in the discrete-time domain if the following values are allocated to the variable parameters $K = 3$, $T_1 = 0.4141$ sec and $T_2 = 0.6172$ sec, resulting in equation (34).

$$G_{032}(s) = \frac{3}{(1 + 0.4141s)(1 + 0.6172s)(1 + 0.8s)} \approx \frac{3}{0.192s^3 + 1.04s^2 + 1.8s + 1} \quad (34)$$

The system instability within the space D(1) is explored in the discrete-time domain if $K = 20$, $T_1 = 0.4141$ sec and $T_2 = 0.6172$ sec, resulting in equation (35).

$$G_{033}(s) = \frac{20}{(1 + 0.4141s)(1 + 0.6172s)(1 + 0.8s)} \approx \frac{20}{0.192s^3 + 1.04s^2 + 1.8s + 1} \quad (35)$$

Then the system's Nyquist and Bode stability assessment in the discrete-time domain is achieved by the code:

```
>> Go31=tf([0 9.034],[0.192 1.04 1.8 1])
Transfer function:
          9.034
-----
0.192 s^3 + 1.04 s^2 + 1.8 s + 1
>> God31 = c2d(Go31,0.05,'tustin')
Transfer function:
0.0006417 z^3 + 0.001925 z^2 + 0.001925 z + 0.0006417
-----
z^3 - 2.742 z^2 + 2.505 z - 0.7626
>> Go32=tf([0 3],[0.192 1.04 1.8 1])
Transfer function:
          3
-----
0.192 s^3 + 1.04 s^2 + 1.8 s + 1
>> God32 = c2d(Go32,0.05,'tustin')
Transfer function:
0.0002139 z^3 + 0.0006417 z^2 + 0.0006417 z + 0.0002139
-----
z^3 - 2.742 z^2 + 2.505 z - 0.7626
>> Go33=tf([0 20],[0.192 1.04 1.8 1])
Transfer function:
          20
-----
0.192 s^3 + 1.04 s^2 + 1.8 s + 1
>> God33 = c2d(Go33,0.05,'tustin')
Transfer function:
0.001426 z^3 + 0.004278 z^2 + 0.004278 z + 0.001426
-----
z^3 - 2.742 z^2 + 2.505 z - 0.7626
Sampling time: 0.05
>> nyquist(God31,God32,God33)
>> margin(God31)
GM = -0.224 dB, PM = -0.921 deg
>> margin(God32)
GM = 9.3 dB, PM = 45 deg
>> margin(God33)
GM = -7.18 dB, PM = -22.8 deg
```

Nyquist Diagrams Proving the D-Partitioning for Cases of Marginal, Stable and Unstable System with Three Variables

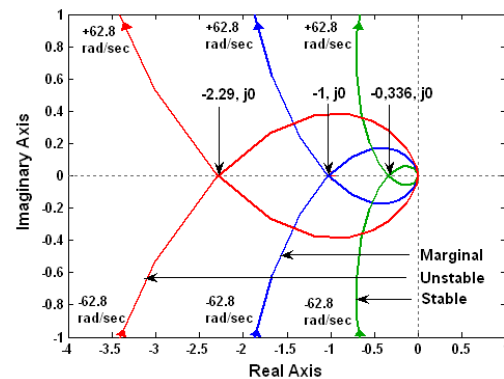


Figure 15. Prove of the D-Partitioning Analysis – Cases of Marginal, Stable and Unstable System with Three Variable Parameters



Applying the Nyquist stability criterion for the cases of marginal, stable and unstable system with three variable parameters, prove exactly the results from the Advanced D-Partitioning in the discrete time-domain.

The system's marginal case is also confirmed with the aid of the Bode stability criterion as shown in Figure 16.

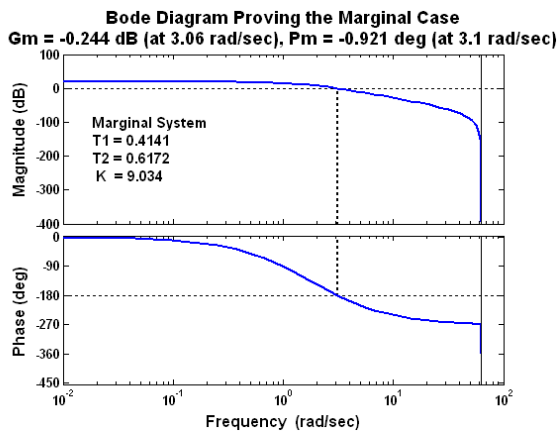


Figure 16. Prove of the D-Partitioning Analysis – Case Marginal System (Case $T_1 = 0.4141$ sec, $T_2 = 0.6172$ sec and $K = 9.034$)

If the variable parameters are $K = 9.034$, $T_1 = 0.4141$ sec and $T_2 = 0.6172$ sec, the results for the digital system gain margin is $GM = -0.244$ dB ≈ 0 dB and the system phase margin is $PM = -0.921^\circ \approx 0^\circ$, as seen in the Bode diagrams in Figure 16. This proves that the three variable parameters have simultaneously marginal values.

The system stability within the space $D(0)$ is explored in the discrete-time domain if the following values are allocated to the variable parameters $K = 3$, $T_1 = 0.4141$ sec and $T_2 = 0.6172$ sec. As seen in the code, the positive values of the gain and phase margins ($GM = 9.3$ dB and $PM = 45^\circ$) prove that if the system's gain is reduced to $K = 3$, while keeping the values of the time constants, the digital control system will operate in the space of stability $D(0)$. Similar result can be achieved if any of the two time constant values is reduced.

The system instability within the space $D(1)$ is explored in the discrete-time domain if $K = 20$, $T_1 = 0.4141$ sec and $T_2 = 0.6172$ sec. From the code above, the negative values of the gain and phase margins ($GM = -7.18$ dB and $PM = -22.8^\circ$) prove that if the system's gain is increased to $K = 20$, while keeping the values of the time constants, the digital control system operates in the space of instability $D(1)$. Similar result can be achieved if any of the two time constant values is increased.

9. Conclusion

The main contribution of this research is the achieved further development of the Advanced D-Partitioning method in its application for the analysis of digital control systems with multivariable parameters. Cases of systems with one, two or three simultaneously variable parameters are illustrated. There is a close match between the discrete and the continuous-time system performance if the Euler's approximation is observed.

The successful application of the analysis method in the discrete-time domain, by employing the Bilinear Tustin

Transform, demonstrates that the basic concept of the Advanced D-Partitioning is relevant and valid for digital control systems as well. The control system's discretization, based on the Bilinear Tustin Transform is providing the most precise results.

When the Advanced D-Partitioning is compared with the Nyquist and the Bode stability criteria, even if both these methods are applicable for continuous and for digital control systems, their disadvantage is that they cannot directly establish the marginal value of a variable parameter. In this case, they can be used only to confirm the results from the Advanced D-Partitioning method.

The research further contributes to knowledge, since the Advanced D-Partitioning is the only method of stability analysis that in terms of its results illustrates graphically the regions of stability and instability in case of system's multivariable parameters. It was demonstrated that any point of the D-partitioning curve or D-partitioning plane, represents the simultaneous marginal values of the two or three variable parameters. This is a unique innovation in the theory of control systems stability analysis.

10. References

- [1] Neimark Y., D-partition and Robust Stability, Computational Mathematics and Modeling, Vol. 9(Issue 2), pp. 160-166, 2006.
- [2] Yanev K. M., Application of the Method of D-Partitioning for Stability of Control Systems with Variable Parameters, Botswana Journal of Technology, Vol.16, Number 1, pp.51-58, 2007.
- [3] Yanev, K.M., Anderson G., Application of the D-partitioning for Analysis and Design of a Robust Photovoltaic Solar Tracker System, International Journal of Energy Systems, Computers and Control, Volume 2, No. 1, ISSN: 0976-6782, pp. 43–54, 2011.
- [4] Yanev K.M., D-Partitioning Analysis of Digital Control Systems by Applying the Bilinear Tustin Approximation, International Review of Automatic Control (IREACO), Napoli, Italy, ISSN: 1974-6059, Vol. 7, N. 6, pp. 517-523, 2014.
- [5] Yanev K.M., "Design and Analysis of Optimal Control System Performance", International Journal of Innovative Science, Engineering & Technology (IJSET), ISSN: 2348-7968, Vol. 4, Issue 2, pp. 232-239, 2017.
- [6] Yanev K. M., Stability of Control Systems with Two Variable Parameters, Proceedings of the Second IASTED Africa Conference on Modelling and Simulation, ISBN 978-0-88986-763-5, pp.106-111, 2008.
- [7] Yanev K. M, Anderson G., Multivariable System's Parameters Interaction and Robust Control Design, Journal of International Review of Automatic Control, Vol. 4, N.2 pp.180-190, 2011.
- [8] Tustin with Frequency Prewarping. MathWorks R2014b Documentation. Retrieved Oct. 17, 2014, <http://radio.feld.cvut.cz/matlab/toolbox/control/ma nipmod/ltiops20.html>



- [9] Continuous-Discrete Conversion Methods. MathWorks R2014b Documentation. Retrieved October 17, 2016, <http://www.mathworks.com/help/control/ug/continuous-discrete-conversionmethods.html#bs78nig-1>
- [10] Euler Method, Retrieved January 17, 2016 from http://en.wikipedia.org/wiki/Euler_method.
- [11] Continuous-Discrete Conversion Methods. Retrieved November 17, 2016, <http://www.mathworks.com/help/control/ug/continuous-discrete-conversion-methods.html#bs78nig-1>
- [12] Yanev K.M., Masupe S., "Robust Design and Efficiency in Case of Parameters Uncertainties, Disturbances and Noise", International Review of Automatic Control, Italy, ISSN: 1974-6059, Vol. 5, N. 6, pp. 860-867, 2012
- [13] Yanev K.M, Anderson G., Achieving Robustness in Control Systems with Variable Time-Constants, International Journal of Energy Systems, Computers and Control, Vol. 1, Number 1, ISSN: 0976-6782, pp.1-14, 2010.
- [14] Driels M., Linear control System Engineering, McGraw-Hill International Inc., pp.145-222, 2006.
- [15] Phillips C., Digital Control System, Prentice- Hall International Inc., pp.125-403, 2005.
- [16] Yanev K.M., Anderson G.O., Masupe S., "Strategy for Analysis and Design of Digital Robust Control Systems", International Congress for Global Science and Technology, ICGST Journal of Automatic Control and System Engineering, Delaware, USA, Volume 12, Issue 1, ISSN: 1687-4811, pp. 37-44, 2012
- [17] Yanev K.M., Design and Analysis of a Robust Accurate Speed Control System by Applying a Digital Compensator, ICGST Journal of Automatic Control and System Engineering, Delaware, USA, Volume 16, Issue 1, ISSN: 1687-4811, pp. 27-36, 2016.

Biographies



Prof. Kamen Yanev has a M.Sc. in Control Systems Engineering. His Ph.D. is in the area of Systems with Variable Parameters and Robust Control Design. He started his academic career in 1974, following a number of academic promotions and being involved in considerable research, service and teaching at different Commonwealth Universities around the world. Prof. Yanev worked in a number of outstanding universities in Europe and in Africa, in countries like Nigeria, Zimbabwe and Botswana. Currently he works as Associate Professor in Control and Instrumentation Engineering at the Department of Electrical Engineering at University of Botswana.

His major research is in the field of Control Systems Engineering as well as in the subjects of Electronics and Instrumentation. He has more than 118 publications in international journals and conference proceedings. Most of his latest publications and current research interests are in the field of Electronics, Analysis of Control Systems with Variable Parameters and Robust Control Design.

Prof. Yanev is interacting with Saint-Cyr French Military Academy, supervising Military Academy students.

He is a member of the Academic Community of International Congress for Global Science and Technology (ICGST), a member of the Instrumentation, Systems and Automation Society (ISA), a member of the Institution of Electrical and Electronic Engineering (IEEE) and the Botswana Institute of Engineers (BIE).

Prof. Yanev is also a member of the Editorial Board of the International Journal of Energy Systems, Computers and Control (IJESCC), a member of the Editorial Board of the Journal of Multidisciplinary Engineering Science and Technology (JMEST), Germany, a reviewer for ACTA Press, Canada.







Multi-Sensor Fusion Indoor Security Based on Interval Probability, Extended Kalman Filter and Fuzzy Rules Pruning

Ali Alqudaihi, Mohamed Zohdy Ph.D. Hoda Abdel-Aty-Zohdy Ph.D.

Department of Electrical and Computer Engineering, Oakland University, Rochester, MI, USA
 aaalquda@oakland.edu, zohdyma@oakland.edu, zohdyhsa@oakland.edu

Abstract

In this paper, noisy heterogeneous multi-sensors, namely Infrared, Active Sonar and Pressure sensors are used for indoor security monitoring. A new approach of enhancing the measurement estimation applying an interval probability to the Extended Kalman Filter process and measurement covariances. A total of four different combinations of the covariances were examined by the addition of upper and lower distributions bounds which helped to find the finest combination that keeps the covariance matrices symmetric and positive definite. In addition, Fuzzy-Logic I rules pruning methods and their applications in security monitoring systems and decision-making rules are addressed and discussed.

Keywords: Multi-Sensor Data Fusion, Interval Probability, Upper/Lower Probability Distribution, Extended Kalman Filter, Measurements Fusion, State Vector Fusion, Fuzzy-Logic I, Rule Pruning.

Nomenclature

DF	Data Fusion
PR	Pressure Sensor
AS	Active Sonar
IR	Infrared Sensor
EKF	Extended Kalman Filter
IP	Interval Probability
PUB	Probability Upper-Bound
PLB	Probability Lower-Bound
FL I	Fuzzy Logic I
SD	Suspiciousness Degree

1. Introduction

Security monitoring systems has been a hot topic for a long time. The aid of utilizing multi-sensory data systems has led to a more reliable tracking and estimation results; which wouldn't be possible if a single sensor's technique is applied especially if users/consumers main concern is accuracy and measurements reliability. In the area of monitoring valuable assets, many areas increased the security devices demand since the ways of breaking-ins has developed as well. Robustness of the system can be enhanced by fusing

multiple sensory data which will make the system less susceptible against the interference that may cause untrustworthy measurements.

The Kalman filter (KF) has found wide application in different controllable systems for identification purposes. The principle theory utilized in these types of filters is the Markov Model that assumes that there exist linear dependencies between systems and Gaussian noise [8].

The importance of multisensory fusion applications has necessitated for increased research about it [2]. As sensory organs are to human, sensors are to systems. They collect information about the system's environment [1] Multisensor fusion involves combining data received from sensors. It could be a single sensor taking data of various parameters or multiple sensors within the multisensor fusion system. The information often obtained is much better and satisfactory than that obtained if the sensors are left to operate independently.

The paper organization is as follows: Section (2) surveys the research's related works and what have been done to the associated case studies. Section (3) reviews the problem motivation and formulation. Sensor Fusion definition, applications, levels, and limitations are discussed in Section (4). Section (5) introduces the Extended Kalman Filter equations and its ongoing cycle. Section (6) provides the theory of interval probability while Section (7) explains how the EKF states and measurements errors are quantified applying the IP. Section (8) discusses the rules pruning methods and how it can reduce Fuzzy Logic I number of rules. Finally, Sections (9), (10) and (11) shows the results of the presented approaches, a conclusion and a reference of sources used in the research consequently.

2. Related Work Survey

Different scholars have in the past utilized the probability intervals in the to extend the Kalman filtering mechanism. Zohdy, Khan and Benedict [12] attempt to explain the use of interval probability in the Kalman filtering mechanism by conducting a Kalman estimation of two filters with diverse gains $K1$, noise covariance Q and a constant KQ . For the first instance, a noise covariance of $Q=10$ is applied to a Kalman filter having an associated gain of $K1=1$ to provide a



constant $kQ = 10$. A second filter utilizes similar parameters as the first but the filter gain is changed. The variation in the filter parameters of the first and second Kalman filters in this example average at 0.08 with the high confidence level being indicated by 0.15 and the low confidence level being indicated by 0.04 [12]. Zohdy, Khan and Benedict's explanations could be utilized in the exploration how the interval probability could be applied to measurement error covariance, and system error covariance as the high support (the alignment of poor measurements) has the capability of lowering the gains for the proceeding iterations after moving the estimator further from the actual state. Consequently, the error would be magnified.

Another work by Hu, Want, Cheng and Zhong is considering the GIP to be applied to system errors and measurement errors. Their method is to differentiate between the state errors and observation errors with random by estimating the bounds of both errors where the EKF can find which error values between those bounds to be chosen [8]. Their proposed method can fine tune the EKF better with GIP compared to the classical EKF.

3. Motivation and Problem Formulation

The security monitoring system with an innovative Fuzzified Extended Kalman Filter and Fuzzy Logic I for decision making is explained thoroughly in a separate research project [1st paper]. The major objective of the system is to track moving agents with three heterogeneous sensors namely: Active Sonar (AS), Infrared (IR), and Pressure Sensitive Floor (PR) sensors. Once the moving agents' tracks are monitored, the security system detect and recognize their intentions toward any valuable/monitored asset in the area under surveillance. The sensors measurements, which are heterogeneous as they provide independent measurement types and homogenous as each moving agent is tracked with three of the same type of sensors, data are filtered and fused to obtain the most accurate distance from those assets and the corresponding weights of each agent. The sensors data are independent and don't communicate with each other; however, the fusion techniques are used to mix their outputs and filter them to minimize any disturbance noises from both the system and the sensors. This research is focusing on applying interval probability theory to modify and enhance the covariances of sensor outputs and states. The results of the improved covariances are compared to the hyper-fuzzified EKF method that was applied in [1st paper]. Moreover, the moving agents' different interaction possibilities are studied and compared which can help to understand how the system can identify how multiple agents communicate with each other. Lastly, rule pruning techniques are manipulated to reduce and/or eliminate any low weight, redundant or unused rules in the decision-making process that utilize Fuzzy Logic I.

4. Sensor Fusion

Sensor Fusion refers to the combination of data drawn from physical sensory data and transforms it to a better form that would enable the use of these sources individually. It could be a single sensor taking data of various parameters or multiple sensors within the multisensor fusion system. Sensor fusion blocks transform received data from the source to a representation that can be fed into the control application thus finishing the process of multisensor integration [2].

The multisensory system has a permanent database that keeps all data thus keeps the system functioning as usual even when there is a partial failure. The data received from each sensor compliments each other therefore limiting the margins of error in interpretation by the controlling application. A multisensory fusion system employs optical and ultrasonic sensors, which means the information data cannot be altered.

The system provides accurate and confirmed measurements. This increase the confidence of the system. When a sensor takes one measurement, it is confirmed by all other sensors working in a similar domain. Also, the accuracy (resolution) is improved because of fusion of the many independent properties results to a better value as compared to the measurement made by a single sensor.

A further benefit of sensor fusion technology is that it helps in reduction of complexity of a system. This is because it can handle a huge amount of data streams that may be ambiguous, imprecise or even incomplete.

Multisensory fusion systems are gaining fame because of their advantages. The system can be reliable since the characteristic intrinsic redundancy can help the system function even in the case of partial failure. The data received from each sensor compliments each other therefore limiting the margins of error in interpretation by the controlling application.

Multisensor Fusion has many applications. Its integration has been useful majorly in security systems. It involves multilevel surveillance and multi-detection systems. Multilevel surveillance structures include active detection modules, passive detection modules, image systems, supervised computers and intelligent homes. It includes motion detectors such as passive infrared, microwave, ultrasonic sensors and vibration sensors and particular types of cameras for images [1]. Security monitoring systems also use databases, human input, and the priori information.

Data is first collected by the security monitoring system and loaded for the fusion process then the data is refined in the object refining process to make it easily identifiable. The refined data is then assessed by analyzing various data collected in the first process to reduce its vulnerability and then stored in a database. Process refinement monitors real-time constraints and can relocate sensors and sources to complete a given task. Finally, results are relayed to the user in the form of light or sound.

In robotics, it increases flexibility and productivity by doing various operations in the industry such as handling of materials, assembly, inspection, and fabrication. It is also



used in military applications such as intelligence collection and analysis, assessment, defense mechanisms et cetera [1]. Multisensor fusion is helpful in remote sensing in the fields of Agriculture, weather forecasting, and climate. Transportation systems such as intelligent cars and other advanced biomedical applications use multisensory fusion [2].

A. Levels of Multisensor Fusion

1. Security monitoring systems are a major application of multisensor fusion. It involves multilevel surveillance and multi-detection systems. Multilevel surveillance structures include active detection modules, passive detection modules, image systems, supervised computers and intelligent homes. It includes motion detectors such as passive infrared, microwave, ultrasonic sensors and vibration sensors and particular types of cameras for images. Security monitoring systems also use databases, human input, and the priori information. This data then goes through the following process:
2. In source preprocessing, also termed as subject-object assessment, data collected by the security monitoring system is allocated to various processes. The main aim of this is to lower the load for the fusion process.
3. Object refinement (object assessment process): It is also labeled as level (1). This stage involves object alignment; changed into similar units and based on a common reference frame by employing the various data classification techniques. It makes the data easily identifiable. For instance, it will enable the security monitoring system to identify a moving object as either a person or animal.
4. Situation refinement (assessment): It involves analyzing the various classes of data from level (1). It then relates the objects to observed events to establish their correlation. Often labeled as level (2)
5. Threat refinement: This is also termed as impact assessment. The security monitoring system uses priori information and predictions to deduce vulnerabilities and when to execute an operation. It also involves database management. Often labeled as level (3)
6. Process refinement: It monitors real-time constraints and can relocate sensors and sources to make a given task complete.
7. The final process is relaying the results to users. It can be in the form of a signal (sound, light, etc.)

B. Sensor Fusion Limitations

Limited coverage regarding space. Normally, one sensor can only cover a defined region. Given the example of temperature recording of water in a boiler, the estimated temperature does not portray that of the boiler. Only the water.

Limits measurements. Some sensors require a specific set up time to work and convey measurements [3]. This limits the maximum number of measurements that can be taken. In this

case, most measurements are gotten from individual sensors thus resulting to imprecision as earlier discussed.

5. Extended Kalman Filter

Extended Kalman Filter (EKF) is a nonlinear form of the well-known KF that was founded by and named after Rudolf Kalman in 1960. The EKF is a recursive estimation algorithm consist of two iterative phases which are the Update/Prediction phase and the Estimation/Correction phase. The filter only needs the most recent measurements to predict or estimate the following state of the system which makes EKF works faster than the other state estimators.

For a discrete nonlinear stochastic difference equation, the model can be formulated as Equations (1) and (2):

$$\tilde{x}_k = f(\tilde{x}_{k-1}, u_{k-1}) + w_{k-1} \quad (1)$$

$$\tilde{y}_k = h(\tilde{x}_{k-1}) + v_{k-1} \quad (2)$$

Where:

x	system state vector
f	nonlinear system's dynamics
u	control vector
w	system's process noise
y	observation equation
h	measurement function
v	measurement's error covariance

w_{k-1} and v_{k-1} , from Equations (1) and (2), are assumed to be system noise and measurement noise respectively where they are white Gaussian with zero mean.

Figure (1) lists all the equations of the EKF and show the ongoing cycle of prediction and corrections phases.

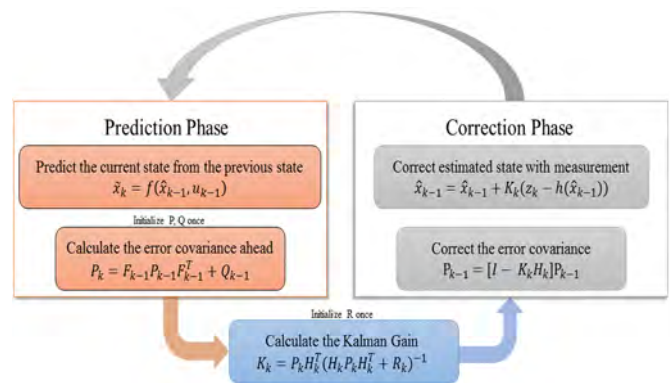


Figure. 1: EKF ongoing cycle and equations

The EKF needs to calculate the Jacobian's matrix, partial derivatives, of the state vector nonlinear elements F in the prediction phase with respect to the states and the Jacobian's matrix of measurements equation elements H in the correction phase with respect to the states again as stated in Equations (3, 4, 5 and 6) correspondingly:

$$F_k = \nabla f_k | \hat{x}_k \quad (3)$$



$$\nabla f_k = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (4)$$

$$H_k = \nabla h_k | \hat{x}_k \quad (5)$$

$$\nabla h_k = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \dots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \dots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \quad (6)$$

6. Interval Probability

Interval probability is a crucial concept utilized in the evaluation of uncertainty [9]. The theory of interval probability bases on a collection of three primary axioms and provides a common ground for the evaluation of the various aspects of uncertainty [10]. Interval probability, also known as the imprecise probability is primarily applied in mechanical systems having conflicting, vague or scarce information regarding probability distribution [11]. The key goal of this research paper is to analyze the extended Kalman filter based on interval probability and the application of the imprecise probability to measurement error covariance and system error covariance.

The Kalman filter is a typical illustration of the applicability of the interval probability. Different scholars have in the past utilized the probability intervals in the to extend the Kalman filtering mechanism.

Robustness plays a crucial role in the Kalman filtering mechanism. The a priori and a posteriori error covariance matrices, $P_{k|k-1}$ and $P_{k|k}$ consequently, of the filter must be kept positive definite and symmetric for it to work excellently [7]. However, round off errors might lead to the violations of the two conditions. Therefore, the loss of symmetry must be dealt with to ensure the achievement of positive definite. The primary causes of non-positive definite might be the measurement update or the time-update equations of the filter [7].

$$P_k = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \quad (7)$$

Equation (7) is considered as the addition of the quantities that are both positive-definite result in a positive definite. The second equation (Equation 8) to be considered is that the error covariance matrix of the measurement update:

$$P_{k-1} = [I - K_k H_k] P_{k-1} \quad (8)$$

The equation yields a positive definite with several round off errors due to the subtraction operation.

7. Error Quantification

Bounds play a crucial role in error quantification. An interval probability that combines general intervals and aspects of probability is used in an extended Kalman filter to quantify irreducible and reducible quantities simultaneously. The

inclusion of distributions between interval probability upper and lower bounds (Equation 9) assist in quantifying systemic errors in Kalman filters thereby assisting designers to eliminate errors [11].

$$([\underline{x}; \bar{x}]): \underline{x}, \bar{x} \in \mathbb{R} \quad (9)$$

The proposed method in this paper is to find the maximum and minimum Probability Interval of the system noise error covariance Q and the measurement noise error covariance R . Both Q and R upper and lower bounds (Equations 10 and 11) represents the highest and lowest interval that can be reached without causing EKF estimations to converge.

$$([\underline{Q}; \bar{Q}]): \underline{Q}, \bar{Q} \in \mathbb{R}, \underline{Q} \geq 0, \bar{Q} \geq 0 \quad (10)$$

$$([\underline{R}; \bar{R}]): \underline{R}, \bar{R} \in \mathbb{R}, \underline{R} \geq 0, \bar{R} \geq 0 \quad (11)$$

Where:

$\underline{Q}$	lower bound of the system error covariance
$\bar{Q}$	upper bound of the system error covariance
$\underline{R}$	lower bound of the measurement error covariance
$\bar{R}$	upper bound of the measurement error covariance

A combination of all possibilities of $\underline{Q}$, $\bar{Q}$, $\underline{R}$ and $\bar{R}$ (Table 1) has been evaluated and compared in such a way that a priori error covariance matrix P_k and a posteriori error covariance matrix P_{k-1} don't lose symmetry and achieve positive definite status. By keeping both P_k and P_{k-1} positive definite the Interval Probability EKF can perform better and become robust to the different noises added by the system states and the measurements error covariances. The impact of the variations of Q and R on the P_k and P_{k-1} is applied to the proposed multi-sensor fusion security monitoring system that was discussed previously in [19] where a Hyper-Fuzzy EKF was developed and investigated to enhance the capabilities of the normal EKF to handle more uncertainty and randomness of the errors obtained from the system and the sensors [19].

The measurement error covariance R have two noises which are r and θ that represent the distance and the angle variances of each sensor as in Equation (12).

$$R = \begin{bmatrix} r_{var} & 0 \\ 0 & \theta_{var} \end{bmatrix} \quad (12)$$

Whereas, Q is the process noise matrix that is diagonal to correspond with the system states as shown in Equation (13).

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 \end{bmatrix} \quad (13)$$



Combination Number	System Error Covariance	Measurements Error Covariance
1	$\underline{Q}$	$\underline{R}$
2	$\underline{Q}$	$\bar{R}$
3	$\bar{Q}$	$\bar{R}$
4	$\bar{Q}$	$\underline{R}$

Table 1: Tested combinations of R and Q covariances

Equation (12) explains the system dynamic model that contain the position, velocity and acceleration on x-axis and y-axis of the moving agents. [19].

$$\begin{bmatrix} \tilde{x}_k(1) \\ \tilde{x}_k(2) \\ \tilde{x}_k(3) \\ \tilde{x}_k(4) \\ \tilde{x}_k(5) \\ \tilde{x}_k(6) \end{bmatrix} = \begin{bmatrix} 1 & dt & \frac{dt^2}{2} & 0 & 0 & 0 \\ 0 & 1 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & \frac{dt^2}{2} \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_{k-1}^x \\ v_{k-1}^x \\ a_{k-1}^x \\ s_{k-1}^y \\ v_{k-1}^y \\ a_{k-1}^y \end{bmatrix} + Bu_{k-1} + \omega_{k-1} \quad (14)$$

Where:

s_k is the position of the tracked agent.

v_k is the velocity of the tracked agent.

a_k is the acceleration of the tracked agent.

The diagonal elements of the error covariances matrix P_k are the variances of the system states with themselves as they are listed in Table 2.

	P_{s^x}	P_{v^x}	P_{a^x}	P_{s^y}	P_{v^y}	P_{a^y}
P_{s^x}	$P(s^x, s^x)$	$P(s^x, v^x)$	$P(s^x, a^x)$	$P(s^x, s^y)$	$P(s^x, v^y)$	$P(s^x, a^y)$
P_{v^x}	$P(v^x, s^x)$	$P(v^x, v^x)$	$P(v^x, a^x)$	$P(v^x, s^y)$	$P(v^x, v^y)$	$P(v^x, a^y)$
P_{a^x}	$P(a^x, s^x)$	$P(a^x, v^x)$	$P(a^x, a^x)$	$P(a^x, s^y)$	$P(a^x, v^y)$	$P(a^x, a^y)$
P_{s^y}	$P(s^y, s^x)$	$P(s^y, v^x)$	$P(s^y, a^x)$	$P(s^y, s^y)$	$P(s^y, v^y)$	$P(s^y, a^y)$
P_{v^y}	$P(v^y, s^x)$	$P(v^y, v^x)$	$P(v^y, a^x)$	$P(v^y, s^y)$	$P(v^y, v^y)$	$P(v^y, a^y)$
P_{a^y}	$P(a^y, s^x)$	$P(a^y, v^x)$	$P(a^y, a^x)$	$P(a^y, s^y)$	$P(a^y, v^y)$	$P(a^y, a^y)$

Table 2: Error covariance P_k elements variances

For each combination of the probability interval errors R and Q (Figure 2), the error covariance matrix P_k is observed in such way that the best combination can hold this matrix symmetrically and achieve positive-definite elements as the IP EKF iterates. Those results are plotted for comparison purposes for the same moving agent track that was created by Matlab (Figure 3).

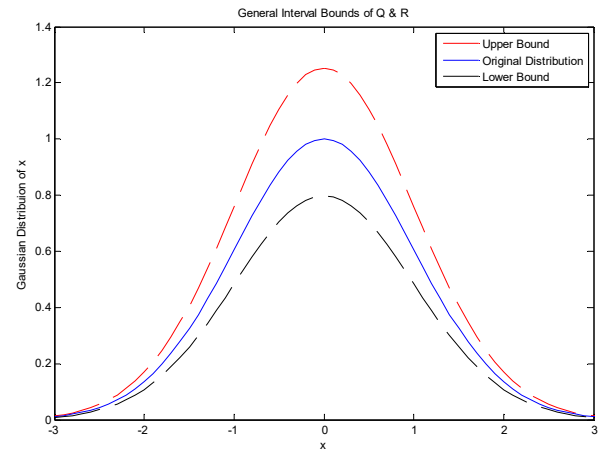


Figure 2: Upper and Lower bounds of IP of Q and R.

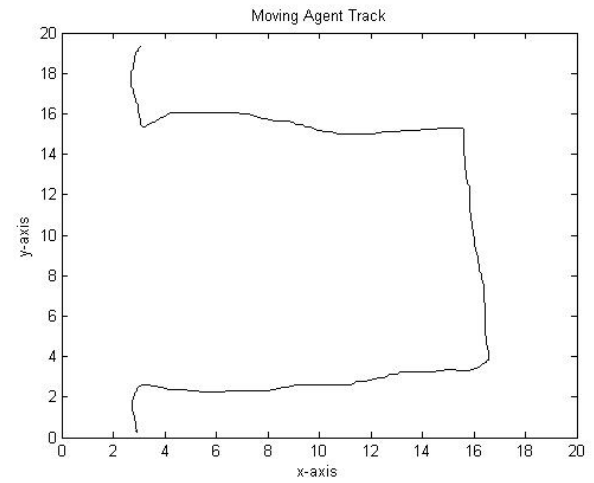
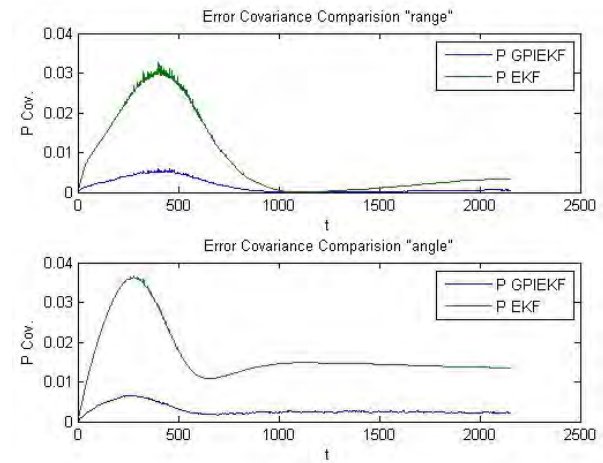
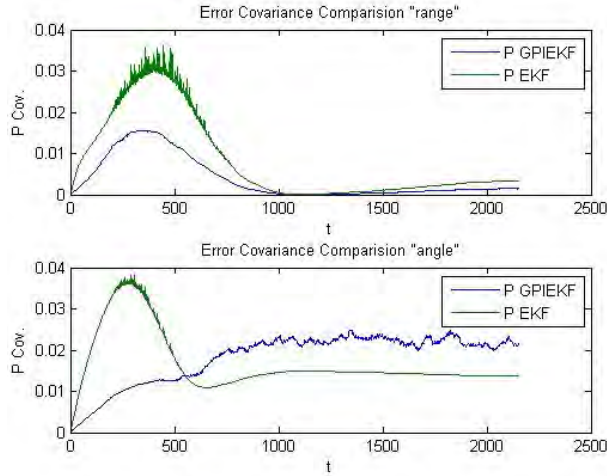
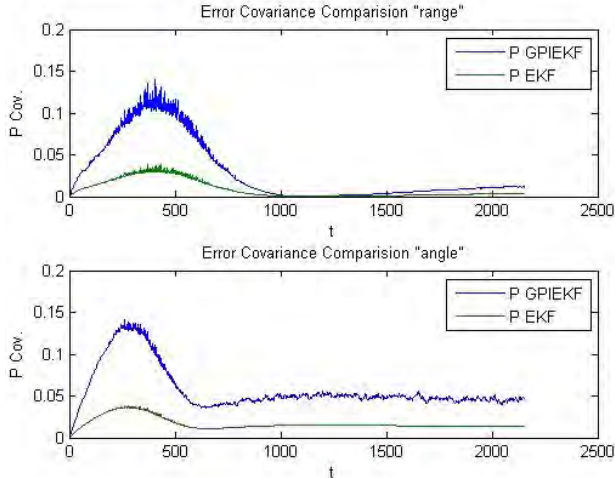
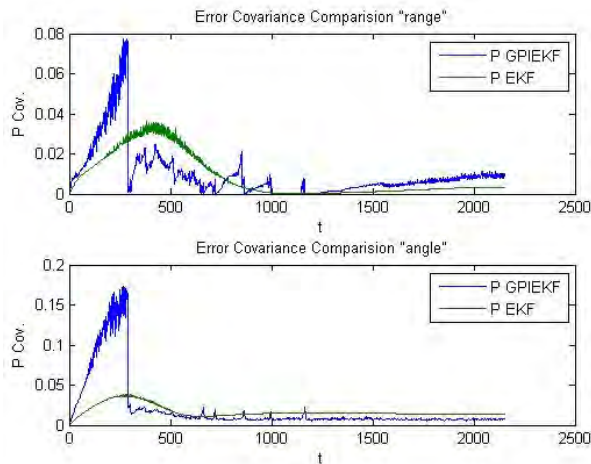


Figure 3: Moving agent track created for testing the GIP.

Combination 1: $\underline{Q}$ (Lower) and $\underline{R}$ (Lower) bounds

Figure 4: Error covariance matrix P with $\underline{Q}$ & $\underline{R}$ 

Combination 2: $\underline{Q}$ (Lower) and $\bar{R}$ (Upper) boundsFigure 5: Error covariance matrix P with $\underline{Q}$ & $\bar{R}$.**Combination 3: $\bar{Q}$ (Upper) and $\bar{R}$ (Upper) bounds**Figure 6: Error covariance matrix P with $\bar{Q}$ & $\bar{R}$.**Combination 4: $\bar{Q}$ (Upper) and $\underline{R}$ (Lower) bounds**Figure 7: Error covariance matrix P with $\bar{Q}$ & $\underline{R}$ **8. Fuzzy Logic I Rules Pruning**

In 1965, Dr. Lotfi Zadeh introduced Fuzzy Logic I Theory and it was presented to compute and handle data with uncertainties to mimic the human being way of making decisions [19]. FL I is able to deal with inaccurate data by applying logical reasoning and make decision based on reasoning even if the data are insufficient [19]. Fuzzy Logic consists of fuzzy sets that have smooth boundaries. The fuzzy sets have a universe of discourse (X) which presents a linear relationship between the membership function $\mu(x)$ and the x-axis [5]. The membership function $\mu(x)$ values ranges between 0 and 1 which cover the discourse elements. Fuzzy set can be presented as:

$$A = \{x, \mu_A(x) | x \in X\} \quad (15)$$

Where A is the fuzzy set, $\mu_A(x)$ is the degree of x in the fuzzy set, and X is the universe of discourse.

Fuzzy logic systems are often used in the applications that require interpretable outputs, and also applications with high level of human interaction like decision making [13]. The advantages of Fuzzy systems are that they provide a simple interaction of the domain expert with the system's designer, and they are also capable of representing the existing uncertainties of human knowledge through linguistic variables [16]. This paper will address the Fuzzy Logic type 1 reduction rules, methods, and their applications in security monitoring systems to decision-making rules.

The sensory fusion approach is a fuzzy reduction method that entails combining sensor signals before using them as input to the Fuzzy Logic Controller (FLC). Using sensory fusion, the reduction will have a lower bound if the variable could be fused. It is also clear that all variables cannot be joined trivially. Each combination has to be explained and reasoned. In practice, variables that are fused include the change of error and the error. The fundamental concepts of the variable structure system are used to determine the change of error fusion and the proposed error. Then the FLC is interpreted as VSS recognizing mode control. For a 2nd order system, the generalized error is obtained from an error signal and its derivative fusion. Therefore, a simpler first order stabilization problem results in S [15].

Hierarchical Structures and Fusion is a reduction technique that was introduced by Raju et al. in these structures the number of rules will increase linearly proportional to the number n of system variables. It is essential but difficult to decide where the variable will be placed in into the hierarchy and it is mostly based on the sensitivity and knowledge of the system analysis. In practice, all variables are categorized according to their importance [15].

A simple rule base reduction algorithm is a reduction technique based on the importance Index. This method is very effective and flexible, particularly for higher order systems. The Importance Index measures the importance of the technique to the controller. The Importance Index is defined as:

$$I_i = \sum_{j=1}^n \beta_i^j \quad (16)$$



When given a controllable cell to control, the rule's contribution to the final crisp controller output is weighted. Then the rule base size is reduced after Incremental Best Estimate Directed Search (IBEDS) converges on the generic rule base [17].

Reduction methods can be applied to a security monitoring system for decision-making rules. Security monitoring control designers can use reduction methods when dealing with a multivariable system. It is essential to reduce the FLC complexity to enhance its synthesis using hierarchy and fusion [18].

Fuzzy techniques can also be used in security monitoring system by using Intrusion Detection Systems (IDS). The Intrusion Detection technology can bring enhanced and flexible detection capability. Intrusion Detection System is a sophisticated decision-making process that involves factors regarding the monitored system. Therefore, fuzzy logic can be applied to reduce false positive of Intrusion Detection Systems [14].

9. Results and Discussion

Heterogenous sensors fusion for a security monitoring system has proven it is superiority over the single sensor system. By combining all data, the tracking accuracy of moving agents of the area under surveillance, the system was able to accurately minimize the measurement errors that were acquired by each single sensor.

The addition of Interval Probability lower and upper bounds for the system and measurement error covariances Q and R to the Extended Kalman Filter has added more controllability and precision to the sensor fusion, allowing the EKF to find the best combination of both error covariances within the bounds of Q and R .

After increasing the uncertainty of the states and the sensors measurements, the 1st and 2nd combinations of Interval Probability of Q and R have shown more stable IPEKF measurement estimation compared to the EKF with no IP.

In all combinations, the estimation of standard EKF diverge after few iterations; however, the IPEKF was able to diverge to the correct estimation in all combinations.

For the 3rd and 4th IP combinations, the IPEKF performance was imprecise which helped to find what Q and R lower and upper bounds combinations that made the estimation not steady.

If IPEKF is compared to the previously presented sensor fusion techniques, namely measurement fusion (MF) and state vector fusion (SVF), that were applied in [19] we can conclude that both techniques can enhance the EKF with more uncertainties presented in the system. IPEKF can be executed faster since the computation is shorter than MF-SVF method. IPEKF and MF-SVF techniques need to be investigated more with different type of applications, and sensors to understand how they differ in improving the EKF. With fine tuning of IPEKF parameters, EKF can estimate the states more accurately than the standard Extended Kalman Filter.

Applying the important index to filter the fuzzy rules of Type I fuzzy logic decision system that was introduced in [19]

reduced the total number of rules which offers a less complex system that can minimize false positive of intrusion detection system.

10. Conclusion

This paper presents an extension of a previous work, namely Hyper-Fuzzy Logic Extended Kalman Filter, applying the same heterogenous sensors, Sensor Fusion techniques and Fuzzy Logic Decision to a security monitoring system. The proposed system utilizes the Interval Probability Theory to expand the range of states and sensors error covariances.

Multi-sensor fusion is very significant and has wide range of applications in indoor security and tracking activities in homes and offices.

The interval probability approach provides an avenue of evaluating the uncertainty in most mechanical systems through the minimization of the measurement error covariance and state or system error covariance. This research defines the various ways in which the Extended Kalman filter could be improved by interval probability and the impact of the variation

Pruning methods can be applied to a security monitoring system for decision-making rules. Security monitoring control designers can use reduction methods when dealing with a multivariable system. It is essential to reduce the FLC complexity to enhance its synthesis using hierarchy and fusion.

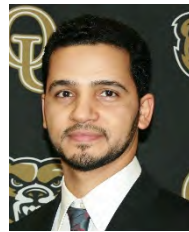
11. References

- [1] Khaleghi, Bahador et al. "Multisensor Data Fusion: A Review of The State-Of-The-Art". Information Fusion, vol 14, no. 1, 2013, pp. 28-44. Elsevier BV.
- [2] Luo, Ren C., and Chih-Chia Chang. "Multisensor Fusion and Integration: A Review on Approaches and Its Applications in Mechatronics". IEEE Transactions on Industrial Informatics, Vol. 8, no. 1, 2012, pp. 49-60. Institute of Electrical and Electronics Engineers (IEEE).
- [3] Luo, Ren C., and Chun Chi Lai. "Multisensor Fusion-Based Concurrent Environment Mapping and Moving Object Detection for Intelligent Service Robotics". IEEE Transactions on Industrial Electronics, Vol. 61, no. 8, 2014, pp. 4043-4051. Institute of Electrical and Electronics Engineers (IEEE).
- [4] T. K. Dakhallallah, M. A. Zohdy, O. M. Salim "Type-2 Fuzzy Kalman Hybrid Application for Dynamic Security Monitoring Systems based on Multiple Sensor Fusion", International Journal on Smart Sensing and Intelligent Systems, Vol.4, No.4, pp. 607-629, 2011.
- [5] O. M. Salim, M. A. Zohdy and H. S. Abdel-Aty-Zohdy, "Applications of hyper-fuzzy modeling and control for bio-inspired systems", 53rd IEEE International Midwest Symposium on Circuits and Systems, Seattle, WA, pp. 169-172, 2010.
- [6] Lundquist, Christian. "Sensor Fusion for Automotive Applications," Linköping University. Linköping, Sweden: LiU-Tryck, Dissertation No. 1409. Year 2011. ISBN 978-91-7393-023-9.



- [7] Higham, Nicholas J. "Computing a nearest symmetric positive semidefinite matrix." *Linear algebra and its applications* 103 (1988): 103-118.
- [8] Hu J, Wang Y, Cheng A, Zhong Z. An Extended Kalman Filtering Mechanism Based on Generalized Interval Probability. *ASME. ASME J. Risk Uncertainty Part B.* 2015;1(3):031002-031002-11. doi:10.1115/1.4030465.
- [9] Wang, Yan. "Generalized fokker-planck equation with generalized interval probability." *Mechanical Systems and Signal Processing* vol.37 no.1 (2013): 92-104.
- [10] Weichselberger, Kurt. "The theory of interval-probability as a unifying concept for uncertainty." *International Journal of Approximate Reasoning* vol.24 no.2-3 (2000): 149-170.
- [11] Xie, Fengyun, Bo Wu, Youmin Hu, Yan Wang, Guangfei Jia, and Yao Cheng. "A generalized interval probability-based optimization method for training generalized hidden Markov model." *Signal Processing* vol.94 (2014): 319-329.
- [12] Zohdy, M. A., Aftab Ali Khan, and Paul Benedict. "Fused multi-sensor data using a Kalman filter modified with interval probability support." *American Control Conference, Proceedings of the 1995. Vol. 5. IEEE, 1995: 3046-3049.*
- [13] Chen, Min-You, and Derek A. Linkens. "Rule-base self-generation and simplification for data-driven fuzzy models." *Fuzzy Systems, 2001. The 10th IEEE International Conference on. Vol. 1. IEEE, (2001): 244-260.*
- [14] Cheung, Yiu-ming, Yue Hao, Licheng Jiao, Yong-Chang Jiao, Jiming Liu, Jianfeng Ma, Yuping Wang, and Hujun Yin. *Computational Intelligence and Security (vol. # 3802): International Conference, Cis 2005, Xi'an, China, December 15-19, 2005, Proceedings, Part Ii. Berlin Heidelberg: Springer-Verlag, 2005. Internet resource*
- [15] Lacrose, Veronique, and Andre Titli. "Fusion and hierarchy can help fuzzy logic controller designers." *Fuzzy Systems, 199, Proceedings of the Sixth IEEE International Conference on. Vol. 2. IEEE, (1997): 1159-1165*
- [16] Mosam Patel. "Various Rule Pruning Techniques and Accuracy measures for Fuzzy rules." *International Journal of Application or Innovation in Engineering & Management (IJAIEEM), Volume 2, Issue 12, (December 2013): pp 175-178*
- [17] Song, Fuijun, and Samuel M. Smith. "A simple weight based fuzzy logic controller rule base reduction method." *Systems, Man, and Cybernetics, 2000 IEEE International Conference on. Vol. 5. IEEE, (2000): 3794-3799*
- [18] Zolghadri Jahromi and Taheri. "A proposed method for learning rule weights in fuzzy rule-based classification systems." *Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran, (2007): 449-459*
- [19] Alqudaihi, Ali, Zohdy, Mohamed, Abdel-Aty-Zohdy, Hoda. "Applying Hyper-Fuzzy Extended Kalman Filter to Indoor Security Monitoring." *Automatic Control and System Engineering ACSE, vol. 17, no. 1, May 2017, pp. 41-51.*

Biographies



Ali A. Alqudaihi was born in Dhahran, Saudi Arabia in 1983. He received his B. Sc. degree in Electrical Engineering from Saginaw Valley State University, Saginaw, Michigan in 2009. He received his M. Sc. degree in Electrical Engineering from Oakland University, Rochester Hills, Michigan in 2011. He is currently a Ph. D. student in Electrical Engineering at Oakland University, Rochester Hills, Michigan. His research interests are in the areas of security and monitoring systems, robotics, nonlinear control, nonlinear estimation and prediction, fuzzy logic and decision making.



Mohamed A. Zohdy was born in Caro, Egypt. He received B.Sc. degree in Electrical Engineering, Cairo University, Egypt in 1968 and he received his M. Sc. and Ph. D. degrees in Electrical Engineering from University of Waterloo, Canada in 1974, and 1977 respectively. He worked in various industries: dowty, iron and steel, and spar. He is currently a Professor at Oakland University, Rochester Hills, Michigan, U.S.A. His research interests are in areas of control, estimation, communication, neural networks, fuzzy logic and hybrid systems.



Hoda S. Abdel-Aty-Zohdy was born in Cairo, Egypt. She received B. Sc. degree (with Frist Class Honors), in Electrical Engineering, Cairo University, Egypt and she received her M. Sc. and Ph. D. degrees from University of Waterloo, Ontario, Canada, all in Electrical Engineering. She has founded the Microelectronics System Design Laboratory at Oakland University. She is currently a Professor at Oakland University, Rochester Hills, Michigan, U.S.A. Her research interests are in the areas of intelligent information processing, embedded Neural Networks, Genetic Algorithms, Analog ICs, Electronic Nose and Microelectronics. Dr. Abdel-Aty-Zohdy is a senior member for the Society of Women Engineers.





Round Robin based Scheduling Algorithms, A Comparative Study

Kamal ElDahshan, Afaf Abd Elkader, Nermeen Ghazy

Dept. of mathematics, Computer science Division, Faculty of science, Al-Azhar University
Cairo, Egypt

[dahshan, afaf2azhar]@azhar.edu.eg, [nermeen_ghazy89]@yahoo.com
<http://www.azhar.edu.eg>

Abstract:

Scheduling is the process of allocating processes to the CPU in order to optimize some objective function. There are many algorithms used to schedule processes. The Round Robin (RR) CPU scheduling algorithm is one of these algorithms which is effective in time sharing and real time operating systems. It gives reasonable response time. But it suffers from several disadvantages such as high turnaround time, high waiting time and many context switches. There are large numbers of algorithms proposed to enhance the standard Round Robin algorithm. In this paper we present a survey with results analysis that conclude recommendations for an Enriched Round Robin algorithm that ameliorates the performance of average waiting time and average turnaround time.

Keywords: Round Robin scheduling algorithm (RR), Adaptive Round Robin Scheduling algorithm, Time Quantum (TQ), Dynamic TQ, Round Robin Remaining time algorithm (RRRT), (IRR) improved Round Robin CPU Scheduling Algorithm, (AAAIRR) an improvement on the improved Round Robin CPU scheduling algorithm, (ERR) An Enhanced Round Robin CPU Scheduling Algorithm, (MMRR) Min-Max Dispersion Measure, (IRRVQ) The improved Round Robin CPU scheduling algorithm with varying time quantum, (AMRR) Average Max Round Robin Scheduling Algorithm, Average waiting time, Average turnaround time.

Nomenclature

RR	Round Robin
TQ	Time Quantum
IRR	Improved Round Robin
AAAIRR	An improvement on the improved Round Robin
ERR	An Enhanced Round Robin CPU Scheduling
RRRT	Round Robin Remaining time algorithm
MMRR	Min-Max Dispersion Measure
IRRVQ	The improved Round Robin CPU scheduling algorithm with varying time quantum
AMRR	Average Max Round Robin
AWT	Average waiting time
ATT	Average turnaround time.
CPU	Central processing unit
ERR	Enriched Round Robin

1. Introduction

The Round Robin (RR) CPU scheduling algorithm referred as Standard RR, it is a preemptive algorithm which assigns a time slice called (time quantum TQ) [1] for each process in the ready queue. When the TQ finishes, the current process is preempted and added to the end of the ready queue [2] [3]. RR is commonly used in time sharing and real time operating system [4] [5] because it gives each process a fair share of time to use the CPU and produces low response time [6]. However, it is known that the Standard RR algorithm suffers from several disadvantages; low throughput (the number of processes that are completed per unit of time) [7], high turnaround time (the time between starting and completion) [7], high waiting time (the amount of time that is waiting in the ready queue) [7], and large number of context switches [6]. The most interesting issue with the Round Robin algorithm is the time quantum [8]. Setting a small TQ causes too many context switches which results in a low performance of the CPU [9] [10]. On the other hand, setting the TQ to a large value may cause a poor response time and the RR algorithm tends to work as FCFS [8] [9] [10].

Many algorithms improved the RR algorithm and produce better results than it. Some of these algorithms use static TQ and others use dynamic TQ. This work is a comparative study of Round Robin and its improvements.

The rest of the paper is organized as follows Section (2) focuses on some Static Time Quantum Algorithms Section (3) focuses on some Dynamic Time Quantum Algorithms Section (4) explains the Results and ameliorations Analysis Section (5) discusses the Enriched Round Robin algorithm Section (6) concludes the paper.

2. Static Time Quantum Algorithms

Some algorithms use TQ which is fixed during the execution of the algorithm. TQ may be assumed or calculated.

The following two subsections discuss both algorithms depending on assumed TQ and calculated TQ.



2.1 Description of the Round Robin algorithms with assumed Time quantum

Some researchers assumed a value for the time quantum which must not exceed the maximum burst time. The following three algorithms IRR, AAIRR, ERR [11] [12] [13] used an assumed time quantum to schedule the processes in the ready queue.

2.1.1 Improved Round Robin CPU Scheduling Algorithm (IRR)

The improved Round Robin (IRR) algorithm is similar to Round Robin (RR) with a small improvement [11]. It selects the first process that arrives in the ready queue and allocates it to the CPU for a time interval of up to 1 time quantum. After completion of process's time quantum, the remaining CPU burst time of the currently running process is checked. If the remaining CPU burst time of the currently running process is less than 1 time quantum, the CPU is again allocated to the currently running process for the remaining CPU burst time. After finishing its execution, it is removed from the ready queue and the next process will be selected. Otherwise, the process will be added to the end of the ready queue and the next process will be selected [11]. It gives better performance than RR where reduces the waiting time and turnaround time.

The pseudo code of the IRR CPU scheduling algorithm in the following:

1. START
2. Make a ready queue of the Processes say REQUEST.
3. Do 4, 5 and 6 WHILE queue REQUEST becomes empty.
4. Pick the first process from the ready queue and allocate the CPU to it for a time interval of up to 1 time quantum.
5. If the remaining CPU burst time of the currently running process is less than 1 time quantum then allocate CPU again to the currently running process for remaining CPU burst time. After completion of execution, removed it from the ready queue and go to 3.
6. Remove the currently running process from the ready queue REQUEST and put it at the tail of the ready queue.
7. END

2.1.2 An Additional Improvement Round Robin CPU scheduling algorithm (AAIRR)

It makes an improvement on the improved Round Robin CPU scheduling algorithm. It selects the first process that arrives in the ready queue and allocates it to the CPU for a time interval of up to 1 time quantum. After completion of process's time quantum, the remaining CPU burst time of the currently running process is checked. If the remaining CPU burst time of the currently running process is less than or equal to 1 time quantum, the CPU is again allocated to the currently running process for the remaining CPU burst time and after execution it removed

from the ready queue then it selects the next shortest process.

Otherwise, the process will be added to the end of the ready queue and the next process in the ready queue will be selected [12]. This algorithm reduces the waiting time and turnaround time compared to the standard Round Robin algorithm and Improved Round Robin algorithm.

The pseudo code of the AAIRR CPU scheduling algorithm in the following:

1. START
2. Make a ready queue of the Processes say REQUEST.
3. Do
4. Pick the first process that arrives to the ready queue and allocate the CPU to it for a time interval of up to 1 time quantum.
5. If the remaining CPU burst time of the currently running process is less or equal to 1 time quantum.
6. Reallocate the CPU again to the currently running process for the remaining CPU burst time. After completing the execution, remove it from the ready queue.
7. Otherwise, remove the currently running process from the ready queue REQUEST and put it at the tail of the ready queue.
8. Pick the next shortest process from the ready queue and allocate the CPU to it for a time interval of up to 1 time quantum then go to 5.
9. WHILE queue REQUEST in not empty
10. Calculate Average Waiting Time, Average Turnaround Time and Number of Context Switch.
11. END

2.1.3 An Enhanced Round Robin CPU Scheduling algorithm (ERR)

This algorithm makes a little improvement on the IRR algorithm. It selects the first process of the ready queue to allocate it to the CPU for a time interval of up to 1 time quantum. Then it checks the remaining burst time of the currently running process, if the remaining burst time is less than or equal to 1 time quantum, the current process is again allocated to the CPU. After completing the execution, this process is removed from the ready queue. If the remaining burst time of the currently running process is longer than 1 time quantum, the process will be added to the end of the ready queue [13].

The pseudo code of the AAIRR CPU scheduling algorithm in the following:

1. Start
2. Make a ready queue of the processes
3. Allocate the CPU to the first process of the ready queue for a time interval of up to 1 time quantum.
4. If the remaining burst time of the currently running process is less than or equal to 1 time quantum then
5. Allocate CPU again to the currently running process for the remaining burst time.



6. After completion of execution remove it from the ready queue and go to 3
7. Repeat 3,4 and 5 WHILE ready queue becomes empty
8. Remove the currently running process from the ready queue and put it at the tail of the ready queue.
9. END

1

2.1.4 Experimental data

There are three cases for incoming burst time namely; random, increasing and decreasing for randomly chosen TQ.

Case 1: The burst time of the processes is random

Consider five processes with a random burst time shown in Table 1. Assume that TQ=12ms, we apply the algorithms RR, IRR, AAIRR and ERR [11,12,13,21] to schedule these processes and the resulting Gantt charts for the RR, IRR, AAIRR and ERR algorithms are shown in Figures 1, 2, 3 and 4 respectively.

Table 1. Five processes with random burst time

process	Burst time
P1	24
P2	40
P3	5
P4	12
P5	34

a) RR algorithm

Enter the processes in the ready queue in the order; P1, P2, P3, P4 and P5. Each process is given a TQ= 12ms periodically.

Gantt chart

P1	P2	P3	P4	P5	P1	P2	P5	P2	P5	P2
0	12	24	29	41	53	65	77	89	101	115

Figure 1: Gantt chart for RR Algorithm (Case 1)

Average waiting time= 49.2 ms

Average turnaround time= 72.2ms

b) IRR algorithm

Here the processes are scheduled according to the IRR algorithm with TQ=12.

Gantt chart

P1	P2	P3	P4	P5	P1	P2	P5	P2
0	12	24	29	41	53	65	77	99

Figure 2: Gantt chart for IRR Algorithm (Case 1)

Average waiting time= 46.8 ms

Average turnaround time= 69.8 ms

c) AAIRR algorithm

Here the processes are scheduled according to the AAIRR algorithm with TQ=12

Gantt chart

P1	P3	P4	P5	P2	P5	P2
0	24	29	41	53	65	87

Figure 3: Gantt chart for AAIRR Algorithm (Case 1)

Average waiting time= 36.2ms

Average turnaround time= 59.2ms

d) ERR algorithm

Here the processes are scheduled according to the ERR algorithm with TQ=12

Gantt chart

P1	P2	P3	P4	P5	P2	P5	P2
0	24	36	41	53	65	77	99

Figure 4: Gantt chart for ERR Algorithm (Case 1)

Average waiting time= 43.4 ms

Average turnaround time= 66.4 ms

The following table compares among the four algorithms RR, IRR, AAIRR and ERR with respect to average waiting time and average turnaround time.

Table 2. Comparison among the RR, IRR, AAIRR and ERR Algorithms

Algorithm	Average waiting time	Average turnaround time
RR	49.2	72.2
IRR	46.8	69.8
AAIRR	36.2	59.2
ERR	43.4	66.4

Figures 5, 6 draw the average waiting time and the average turnaround time for the four algorithms RR, IRR, AAIRR and ERR algorithms.

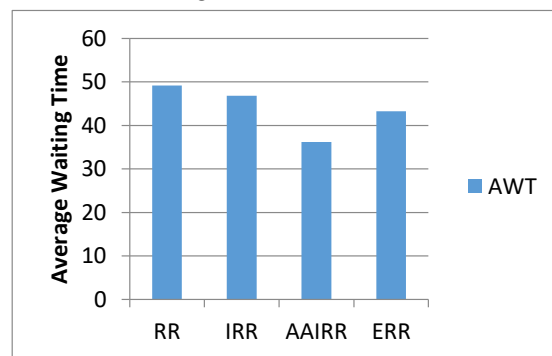


Figure 5: Graph for Average Waiting Time for RR, IRR, AAIRR and ERR algorithms (Case 1)



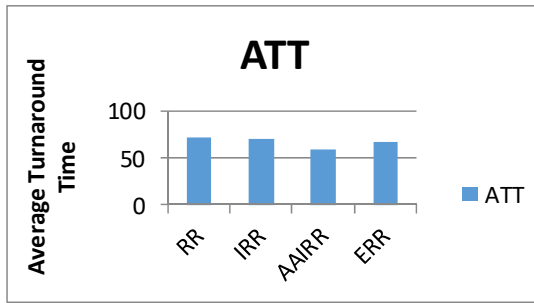


Figure 6: Graph for Average Turnaround Time for RR, IRR, AAIRR and ERR algorithms (Case 1)

From Figures 5, 6 it is noticed that the algorithms IRR, AAIRR and ERR give better results than the standard Round Robin algorithm for the average waiting time and the average turnaround time. By comparing the results of the three algorithms, it is noticed that the ERR algorithm reduces the average waiting time and average turnaround time than the IRR algorithm. However the AAIRR algorithm is the best one of them.

Case 2: the burst time of the processes is in an increasing order

Consider five processes with a randomly increasing order of burst time shown in Table 3 assume that $TQ=10$ ms, we apply the algorithms RR, IRR, AAIRR and ERR to schedule these processes and the resulting Gantt charts for the RR, IRR, AAIRR and ERR algorithms are shown in Figures 7, 8, 9 and 10 respectively.

process	Burst time
P1	7
P2	10
P3	20
P4	26
P5	30

a) RR algorithm

Enter the processes in the ready queue in the order; P1, P2, P3, P4 and P5. Each process is given a $TQ=10$ ms periodically.

Gantt chart

P1	P2	P3	P4	P5	P3	P4	P5	P4	P5	
0	7	17	27	37	47	57	67	77	83	93

Figure 7: Gantt chart for RR Algorithm (Case 2)

Average waiting time= 32.8 ms
Average turnaround time= 51.4ms

b) IRR algorithm

Here the processes are scheduled according to the IRR algorithm with $TQ=10$.

Gantt chart

P1	P2	P3	P4	P5	P3	P4	P5	
0	7	17	27	37	47	57	73	93

Figure 8: Gantt chart for IRR Algorithm (Case 2)

Average waiting time= 30.8 ms
Average turnaround time= 49.4 ms

c) AAIRR algorithm

Here the processes are scheduled according to the AAIRR algorithm with $TQ=10$.

Gantt chart

P1	P2	P3	P4	P5	P4	P5	
0	7	17	37	47	57	73	93

Figure 9: Gantt chart for AAIRR Algorithm (Case 2)

Average waiting time= 26.8ms
Average turnaround time= 45.4ms

d) ERR algorithm

Here the processes are scheduled according to the ERR algorithm with $TQ=10$.

Gantt chart

P1	P2	P3	P4	P5	P4	P5	
0	7	17	37	47	57	73	93

Figure 10: Gantt chart for ERR Algorithm (Case 2)

Average waiting time= 26.8 ms
Average turnaround time= 45.4 ms

The following table compares among the four algorithms RR, IRR, AAIRR and ERR with respect to average waiting time and average turnaround time.

Table 4. Comparison among the RR, IRR, AAIRR and ERR Algorithms

Algorithm	Average waiting time	Average turnaround time
RR	32.8	51.4
IRR	30.8	49.4
AAIRR	26.8	45.4
ERR	26.8	45.4

Figures 7, 8 draw the average waiting time and the average turnaround time for the four algorithms RR, IRR, AAIRR and ERR.



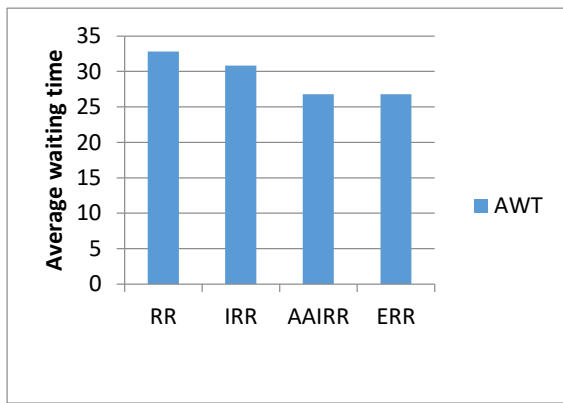


Figure 11: Graph for Average Waiting Time for RR, IRR, AAIRR and ERR algorithms (Case 2)

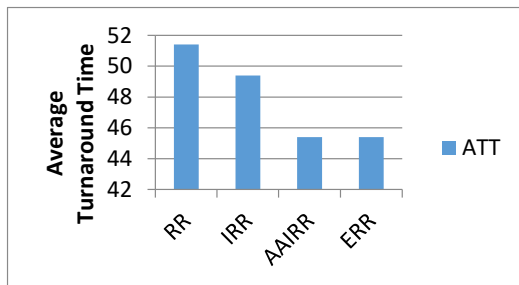


Figure 12: Graph for Average Turnaround Time for RR, IRR, AAIRR and ERR algorithms (Case 2)

From above Figures 11, 12 it is noticed that the algorithms IRR, AAIRR and ERR are doing better than standard Round Robin algorithm for the average waiting time and the average turnaround time. For the other three algorithms, it is noticed that the ERR algorithm gives the same results as the AAIRR algorithm because of the increasing order of the incoming processes. But the IRR algorithm gives high results than both of them. The benefit of AAIRR algorithm which is selecting the next short process to assign it to the CPU which leads to minimize the average waiting time and average turnaround time is not exist.

Case 3: The burst time of the processes is in decreasing order

Consider five processes with a randomly decreasing order of burst time shown in Table 5. Assume that $TQ=15ms$, we apply the algorithms RR, IRR, AAIRR and ERR to schedule these processes and the resulting Gantt charts for the RR, IRR, AAIRR and ERR algorithms are shown in Figures 13, 14, 15 and 16 respectively.

Table 5. Burst time in decreasing order

process	Burst time
P1	45
P2	36
P3	30
P4	18
P5	10

a) RR algorithm

Enter the processes in the ready queue in order; P1, P2, P3, P4 and P5. Each process is given a $TQ=15ms$ periodically.

Gantt chart

P1	P2	P3	P4	P5	P1	P2	P3	P4	P1	P2
0	15	30	45	60	70	85	100	115	133	139

Figure 13: Gantt chart for RR Algorithm (Case 3)

Average waiting time= 87.2 ms

Average turnaround time= 115ms

b) IRR algorithm

Here the processes are scheduled according to the IRR algorithm with $TQ=15$.

Gantt chart

P1	P2	P3	P4	P5	P1	P2	P3	P1	
0	15	30	45	63	73	88	109	124	139

Figure 14: Gantt chart for IRR Algorithm (Case 3)

Average waiting time= 73.8 ms

Average turnaround time= 101.6 ms

c) AAIRR algorithm

Here the processes are scheduled according to the AAIRR algorithm with $TQ=15$.

Gantt chart

DATA							
P1	P5	P4	P3	P2	P1	P2	
0	15	25	43	73	88	118	139

Figure 15: Gantt chart for AAIRR Algorithm (Case 3)

Average waiting time= 51.8ms

Average turnaround time= 79.6ms

d) ERR algorithm

Here the processes are scheduled according to the ERR algorithm with $TQ=15$.

Gantt chart

P1	P2	P3	P4	P5	P1	P2	
0	15	30	60	78	88	118	139

Figure 16: Gantt chart for ERR Algorithm (Case 3)

Average waiting time= 68.8 ms

Average turnaround time= 96.6 ms

The following table compares among the four algorithms RR, IRR, AAIRR and ERR with respect to average waiting time and average turnaround time.



Table 6. Comparison among the RR, IRR, AAIRR and ERR Algorithms

Algorithm	Average waiting time	Average Turnaround time
RR	87.2	115
IRR	73.8	101.6
AAIRR	51.8	79.6
ERR	68.8	96.6

Figures 17, 18 draw the average waiting time and the average turnaround time for the RR, IRR, AAIRR and ERR algorithms.

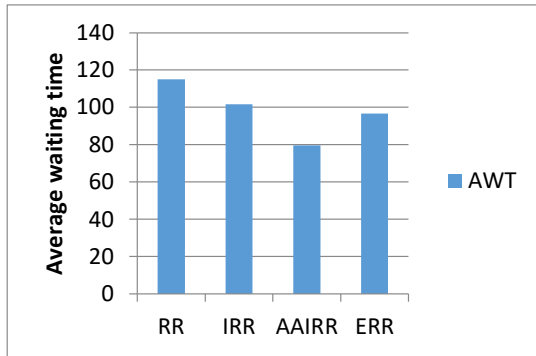


Figure 17: Graph for Average Waiting Time for RR, IRR, AAIRR and ERR algorithms (Case 3)

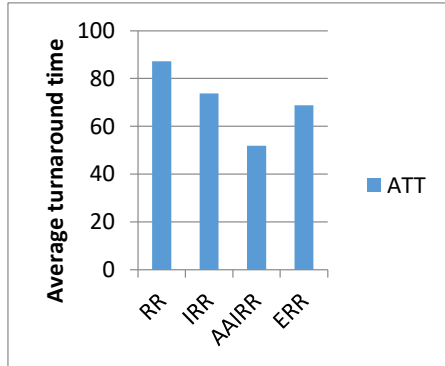


Figure 18: Graph for Average Turnaround Time for RR, IRR, AAIRR and ERR algorithms (Case 3)

From Figures 17, 18 it is noticed that the algorithms IRR, AAIRR and ERR give better results than the standard Round Robin algorithm for the average waiting time and the average turnaround time. By comparing the results of the three algorithms, it is noticed that the ERR algorithm reduces the average waiting time and average turnaround time than the IRR algorithm. However the AAIRR algorithm is the best one of them.

Hence, the AAIRR algorithm is the algorithm which gives the best results in random and decreasing cases. But in the case of increasing order it gives the same results as the ERR algorithm.

2.2 Description of the Round Robin algorithms with calculated time quantum and their Experimental data

2.2.1 Adaptive Round Robin Scheduling Algorithm (Adaptive RR)

This algorithm improved the Round Robin algorithm. It is a “priority scheduling algorithm” based on the burst time of processes. It is composed of two phases. First, the processes are arranged according to the burst time. The highest priority is given to the process with smallest burst time. This approach chooses the smart time slice depending on the number of processes. If the number of processes is even, then the smart time slice will be the average of the burst of all processes. If the number of processes is odd, then the smart time slice is equal to the burst time of the mid-process [14].

Adaptive Round Robin Pseudo code as follow.

1. First of all check ready queue is empty
2. When ready queue is empty then all the processes are assigned into the ready queue.
3. All the processes are rearranged in increasing order that means smaller burst time process get higher priority and larger burst time process get lower priority.
4. While (ready queue != NULL)
5. Calculate smart Time Slice:
If (Number of process%2= 0)
STS = average CPU burst time of all processes
Else
STS = mid process burst time
6. Assign smart time slice to the ith process:
Pi_STs
7. If (i< Number of process) then go to step 6.
8. If a new process is arrived update the ready queue, go to step 2.
9. End of While
10. Calculate average waiting time, turnaround time, context switches and throughput.
11. End

2.2.2 Round Robin Remaining Time Algorithm (RRRT)

This algorithm improved the Adaptive Round Robin Scheduling algorithm. It assumes that all processes arrive at the same time in the ready queue, then they are arranged in an ascending order according to their burst time. TQ is calculated by $\sum p_i / 2n$. If the remaining CPU burst time of the currently running process is less than the TQ, the CPU is again allocated to the currently running process for the remaining CPU burst time. Otherwise, the process will be added to the end of the ready queue [15].



The pseudo code of the Round Robin Remaining Time Algorithm:

1. Assign Process to ready queue.
2. Rearrange all the process in increasing order of their burst time
3. While (ready queue! =NULL)
4. Calculate time quantum = $\sum p_i / 2 * n$
5. if (remaining burst time < time quantum)
Allocate CPU again to the current running process for remaining burst time
Else
Remove the current running process from the ready queue and put it at the tail of the ready queue.
6. If no of process > 0 Go to 5
7. End while
8. Calculate average waiting time, average turnaround time.

2.1.3 Experimental data

There are three cases for incoming burst time; random, increasing and decreasing. The order of the processes affects the results of the Round Robin algorithm since the processes are executed in their given order, whereas in the other algorithms the burst time of the processes is arranged in an ascending order in all cases. The results of the three cases are described below.

Case 1: The burst time of the processes is random

Consider five processes with a random burst time shown in Table 7. We apply the algorithms RR, Adaptive RR and RRRT to schedule these processes and the resulting Gantt charts for the RR, Adaptive RR and RRRT algorithms are shown in Figures 19, 20 and 21 respectively.

Table 7. Five processes with random burst time

process	Burst time
P1	18
P2	38
P3	6
P4	28
P5	21

a) RR algorithm

Enter the processes in the ready queue in the order; P1, P2, P3, P4 and P5. Each process is given a TQ= 14ms periodically.

Gantt chart

P1	P2	P3	P4	P5	P1	P2	P4	P5	P2	
0	14	28	34	48	62	66	80	94	101	111

Figure 19: Gantt chart for RR Algorithm (Case 1)

Average waiting time= 59ms

Average turnaround time= 81.2ms



b) Adaptive RR algorithm

Arrange the processes in the ready queue in an ascending order of their burst time; P3=6, P1=18, P5=21, P4=28 and P2=38. Since the number of processes is odd, the time quantum is set to be the burst time of the mid process. Hence, TQ=21 according to Adaptive RR algorithm.

Gantt chart

P3	P1	P5	P4	P2	P4	P2	
0	6	24	45	66	87	94	111

Figure 20: Gantt chart for Adaptive RR Algorithm (Case 1)

Average waiting time= 33.8 ms

Average turnaround time= 56 ms

c) RRRT algorithm

Arrange the processes in the ready queue according to their given burst time in an ascending order; P3=6, P1=18, P5=21, P4=28 and P2=38. The time quantum is calculated according to the RRRT Algorithm. The resulting TQ is 11ms.

Gantt chart

P3	P1	P5	P4	P2	P4	P2	
0	6	24	45	56	67	84	111

Figure 21: Gantt chart for RRRT Algorithm (Case 1)

Average waiting time= 31.8ms

Average turnaround time= 54ms

The following table compares among the three algorithms RR, Adaptive RR and RRRT with respect to average waiting time and average turnaround time.

Table 8. Comparison among the RR, Adaptive RR and RRRT Algorithms

Algorithm	Average waiting time	Average Turnaround time
RR	59	81.2
Adaptive RR	33.8	56
Round Robin Remaining Time	31.8	54

Figures 22, 23 draw the average waiting time and the average turnaround time for the RR, Adaptive RR and RRRT algorithms.



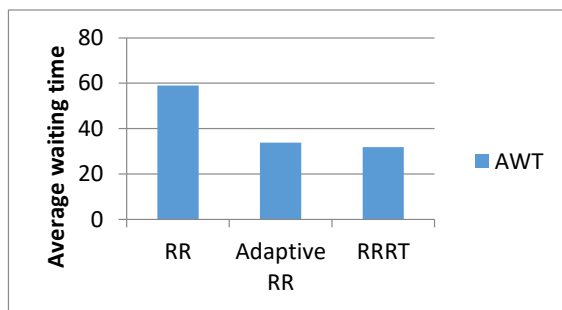


Figure 22: Graph for Average Waiting Time for RR, Adaptive RR and RRRT algorithms (Case 1)

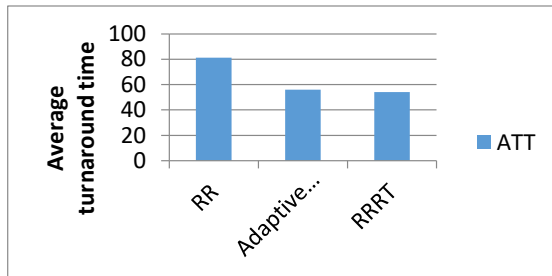


Figure 23: Graph for Average Turnaround Time for RR, Adaptive RR and RRRT algorithms (Case 1)

Case 2: The burst time of the processes is in an increasing order

Consider five processes with a randomly increasing order of burst time shown in Table 9. We apply the algorithms RR, Adaptive RR and RRRT to schedule these processes and the resulting Gantt charts for the RR, Adaptive RR and RRRT algorithms are shown in Figures 24, 25 and 26 respectively.

Table 9. Five processes with random burst time

process	Burst time
P1	15
P2	19
P3	23
P4	47
P5	56

a) RR algorithm

Enter the processes in the ready queue in order; P1, P2, P3, P4 and P5.

Each process is given a TQ= 20ms periodically.

Gantt chart

P1	P2	P3	P4	P5	P3	P4	P5	P4	P5	
0	15	34	54	74	94	97	117	137	144	160

Figure 24: Gantt chart for RR Algorithm (Case 2)

Average waiting time= 58ms

Average turnaround time= 90ms

b) Adaptive RR algorithm

Arrange the processes in the ready queue in an ascending order of their burst time; P1=15, P2=19, P3=23, P4=47 and P5=56. Since the number of processes is odd, the time quantum is set to be the burst time of the mid process. Hence, TQ=23 according to Adaptive RR algorithm.

Gantt chart

P1	P2	P3	P4	P5	P4	P5	P4	P5
0	15	34	57	80	103	126	149	150 160

Figure 25: Gantt chart for Adaptive RR Algorithm (Case 2)

Average waiting time= 51.2 ms

Average turnaround time= 83.2 ms

c) RRRT algorithm

Arrange the processes in the ready queue according to their given burst time in an ascending order; P1=15, P2=19, P3=23, P4=47 and P5=56. The time quantum is calculated according to the RRRT Algorithm. The TQ is calculated as 16ms.

Gantt chart

P1	P2	P3	P4	P5	P4	P5	
0	15	34	57	73	89	120	160

Figure 26: Gantt chart for RRRT Algorithm (Case 2)

Average waiting time= 45.2ms

Average turnaround time= 77.2ms

The following table compares among the three algorithms RR, Adaptive RR and RRRT with respect to average waiting time and average turnaround time.

Table 10. Comparison among the RR, Adaptive RR and RRRT Algorithms

Algorithm	Average waiting time	Average Turnaround time
RR	58	90
Adaptive RR	51.2	83.2
Round Robin Remaining Time	45.2	77.2

Figures 27, 28 draw the comparison of average waiting time and average turnaround time for the RR, Adaptive RR and RRRT algorithms.

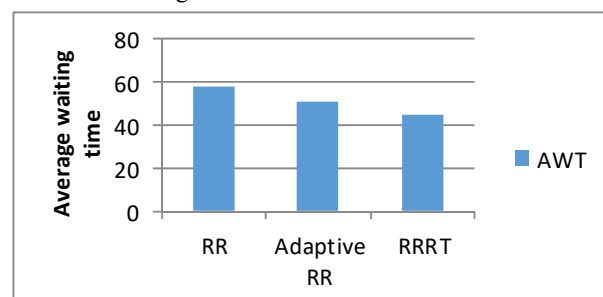


Figure 27: Graph for Average Waiting Time for RR, Adaptive RR and RRRT algorithms (Case 2)



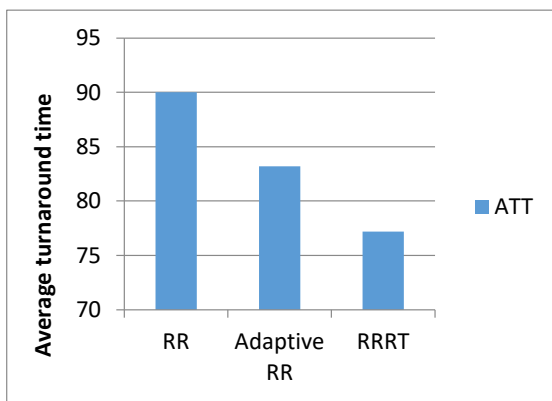


Figure 28: Graph for Average Turnaround Time for RR, Adaptive RR and RRRT algorithms (Case 2)

Case 3: The burst time of the processes is in decreasing order

Consider five processes with a randomly decreasing order of burst time shown in Table 11. We apply the algorithms RR, Adaptive RR and RRRT to schedule these processes and the resulting Gantt charts for the RR, Adaptive RR and RRRT algorithms are shown in Figures 29, 30 and 31 respectively.

Table 11. Burst time in decreasing order

process	Burst time
P1	70
P2	53
P3	26
P4	20
P5	15

a) RR algorithm

Enter the processes in the ready queue in order; P1, P2, P3, P4 and P5.

Each process is given a TQ= 25ms periodically.

Gantt chart

P1	P2	P3	P4	P5	P1	P2	P3	P1	P2
0	25	50	75	95	110	135	160	181	184

Figure 29: Gantt chart for RR Algorithm (Case 3)

Average waiting time= 109.4ms

Average turnaround time= 146.2ms

b) Adaptive RR algorithm

Arrange the processes in the ready queue in an ascending order of their burst time; P5=15, P4=20, P3=26, P2=53 and P1=70. Since the number of processes is odd, the time quantum is set to be the burst time of the mid process. Hence, TQ=26 according to Adaptive RR algorithm.

Gantt chart

P5	P4	P3	P2	P1	P2	P1	P2	P1
0	15	35	61	87	113	139	165	184

Figure 30: Gantt chart for Adaptive RR Algorithm (Case 3)

Average waiting time= 55.4 ms

Average turnaround time= 92.2 ms

c) RRRT algorithm

Arrange the processes in the ready queue according to their given burst time in an ascending order; P5=15, P4=20, P3=26, P2=53 and P1=70. The time quantum is calculated according to the RRRT Algorithm. The TQ is calculated as 18ms.

Gantt chart

P5	P4	P3	P2	P1	P2	P1
0	15	35	61	79	97	132

Figure 31: Gantt chart for RRRT Algorithm (Case 3)

Average waiting time= 48.6ms

Average turnaround time= 85.4ms

The following table compares among the three algorithms RR, Adaptive RR and RRRT with respect to average waiting time and average turnaround time.

Table 12. Comparison among the RR, Adaptive RR and RRRT Algorithms

Algorithm	Average waiting time	Average Turnaround time
RR	109.4	146.2
Adaptive RR	55.4	92.2
Round Robin Remaining Time	48.6	85.4

Figures 32, 33 draw the comparison of average waiting time and average turnaround time for the RR, Adaptive RR and RRRT algorithms.

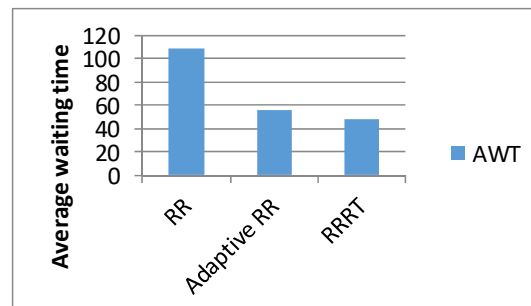


Figure 32: Graph for Average Waiting Time for RR, Adaptive RR and RRRT algorithms (Case 3)



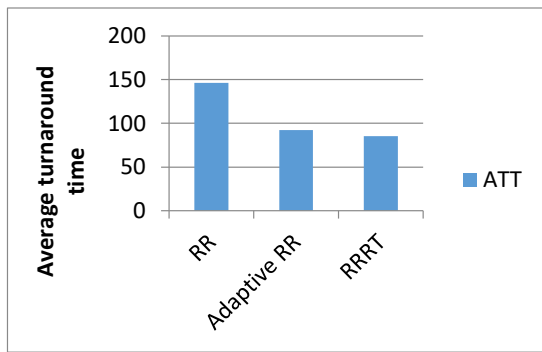


Figure 33: Graph for Average Turnaround Time for RR, Adaptive RR and RRRT algorithms (Case 3)

3. Dynamic TQ

The performance of RR algorithm depends on the fixed Time Quantum [6]. If it assumes a large TQ, then many processes whose burst time less than TQ will be finished during its TQ. Hence the RR algorithm tends to be a FCFS algorithm. On the other hand if TQ is very small, the processes will be interrupted many times. These interruptions increase the overhead of the CPU which leads to a large number of context switches.

A dynamic TQ may changes many times during the execution of the algorithm depending on the current values of the process's burst time [16].

Many algorithms are developed to improve the efficiency of the RR algorithm using dynamic TQ.

The following is a comparison among the MMRR, IRRVQ, AMRR [17] [18] [19] and RR algorithms.

3.1 Min-Max Dispersion Measure (MMRR)

This algorithm depends on dynamic TQ; the processes are sorted in an increasing order of their burst time. It takes TQ as the range of the CPU burst time of all the processes which is the difference between the maximum and minimum values of the processes' burst time. If this range is less than 25 then the next TQ is set to 25, otherwise the next TQ is set to the calculated TQ [7].

Pseudo code for Min-Max Round Robin (MMRR) algorithm is as follow.

1. All the processes present in the ready queue are sorted in ascending order.
//n = number of processes, i = loop variable
2. while (RQ != NULL)
//RQ = Ready Queue
TQ = MAXBT – MINBT
//TQ = Time Quantum
//MAXBT = MAXimum Burst Time
//MINBT = MINimum Burst Time
(Remaining burst time of the processes)
// If one process is there then TQ is equal to BT of itself
3. if (TQ < 25)
set TQ_{new} = 25
else
set TQ_{new} = TQ
end if

4. //Assign TQ to (1 to n) process
for i = 1 to n
{
P_i → TQ_{new}
}
end for
// Assign TQ_{new} to all the available processes.
5. Calculate the remaining burst time of the processes.
6. if (new process is arrived and BT != 0)
go to step 1
else if (new process is not arrived and BT != 0)
go to step 2
else if (new process is arrived and BT == 0)
go to step 1
else
go to step 7
end if
end while
7. Calculate ATT, AWT and CS.
//ATT = Average Turnaround Time
//AWT = Average Waiting Time
//CS = number of Context Switches
8. End

Flowchart of Min-Max Round Robin (MMRR) algorithm

3.2 The improved Round Robin CPU scheduling algorithm with varying time quantum (IRRVQ)

This algorithm associates the features of SJF and RR scheduling algorithms with varying time quantum. First, it arranges all the processes in the ready queue in an ascending order. The processes are allocated to the CPU using RR scheduling with time quantum value equal to the first process's burst time in the ready queue. After each iteration the remaining processes are again arranged in an ascending order and allocated to the CPU using RR scheduling with TQ equal to the burst time of the first process in the ready queue [8].

Pseudo code of the IRRVQ CPU scheduling algorithm is as follow.

1. Make a ready queue RQUEUE of the Processes submitted for execution.
2. DO steps 3 to 9 WHILE queue RQUEUE becomes empty.
3. Arrange the processes in the ready queue REQUEST in the ascending order of their remaining burst time.
4. Set the time quantum value equal to the burst time of first process in the ready queue RQUEUE.
5. Pick the first process from the ready queue RQUEUE and allocate CPU to this process for a time interval of up to 1 time quantum.
6. Remove the currently running process from the ready queue RQUEUE, since it has finished execution and the remaining burst time is zero.



7. REPEAT steps 8 and 9 UNTIL all processes in the ready queue gets the CPU time interval up to 1 time quantum.
8. Pick the next process from the ready queue RQUEUE, and allocate CPU for a time interval of up to 1 time quantum.
9. IF the currently running process has finished execution and the remaining CPU burst time of the currently running process is zero
remove it from the ready queue
ELSE
remove the currently running process from the ready queue RQUEUE and put it at the tail of the ready queue.

3.3 Average Max Round Robin Scheduling Algorithm (AMRR)

This algorithm uses a dynamic TQ to minimize the average waiting time and average turnaround time. All the processes are arranged in an ascending order of their burst time. The TQ is calculated where TQ is equal to (average of burst times + maximum burst time)/2.

AVG = (Summation of Burst Time of all the processes)/
Number of processes
TQ = (AVG + MAXBT)/2

After first iteration the processes are arranged again in an ascending order of their remaining burst time and new TQ is calculated [9].

Pseudo code for Avg Max Round Robin (AMRR) algorithm as follow.

1. All the process in the ready Queue are stored in ascending order
// n= Number of processes // i= Loop variable (counter) //BT= Burst Time
2. While (RQ!=NULL) // RQ=Ready Queue
//TQ=Time Quantum //AVG=Average of all the processes
//MAXBT=Maximum Burst Time AVG= (Sum of BT of all the process)/Number of processes.
//(Use round off function in AVG.)
TQ=(AVG+MAXBT)/2
//(Use round off function in TQ) (Remaining Burst time of the process)
//If one process is there then TQ is equal to BT itself
3. // Assign TQ to (1 to n) processes
for i=0 to n loop
{
 Pi->TQn
}
//TQn=New Time Quantum
end of for
//Assign TQn to all the available processes
4. Calculate the remaining Burst time of the processes.

5. If (new process arrived and BT!=0 or new process is arrived and BT==0)

then go to step1
else if (new process is not arrived and BT!=0)
then go to step 2
else
go to step 6
end of if
end of while

6. Calculate ATT,AWT,CS

//ATT=Average turnaround time
//AWT=Average waiting time //CS=Number of context switch

7. End

3.4 Simulation

Consider five processes with random burst time arrived at time 0 as shown in Table 13. The Gantt charts for the RR, MMRR, IRRVQ and AMRR algorithms are shown in Figures 34, 35, 36 and 37 respectively.

Table 13. Burst time for processes

Process	Burst time
P1	20
P2	55
P3	38
P4	24
P5	40

a) RR algorithm

Enter the processes in the ready queue in order; P1, P2, P3, P4 and P5.

Each process is given a TQ= 22ms periodically.

Gantt chart

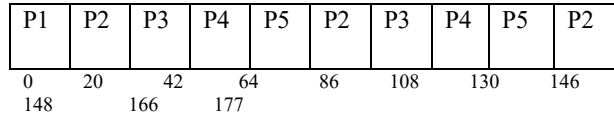


Figure 34: Gantt chart for RR Algorithm

Average waiting time= 109.4ms

Average turnaround time= 146.2ms

b) MMRR algorithm

Arrange all the processes in the ready queue in an ascending order according to their burst time; P1=20, P4=24, P3=38, P5=40 and P2=55. The TQ calculated as the difference between the maximum burst time and the minimum burst time.

Gantt chart

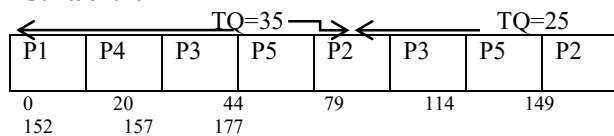


Figure 35: Gantt chart for Adaptive MMRR Algorithm

Average waiting time= 65.8 ms

Average turnaround time= 110 ms



c) IRRVQ algorithm

Arrange all the processes in the ready queue in an ascending order according to their burst time; P1=20, P4=24, P3=38, P5=40 and P2=55. In each iteration, the TQ is equal to the first process's burst time.

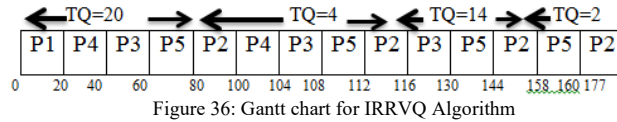
Gantt chart

Figure 36: Gantt chart for IRRVQ Algorithm

Average waiting time= 82.8ms

Average turnaround time= 118.2ms

d) AMRR algorithm

Arrange all the processes in the ready queue in an ascending order according to their given burst time in an ascending order; P1=20, P4=24, P3=38, P5=40 and P2=55. The TQ is the mean of (average of BT + Max BT)

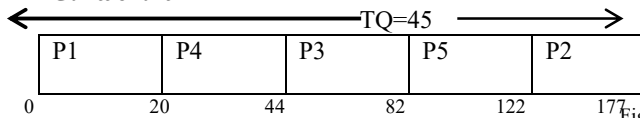
Gantt chart

Figure 37: Gantt chart for AMRR Algorithm

Average waiting time= 53.6ms

Average turnaround time= 89ms

Summarizes the results obtained above with respect to average waiting time and average turnaround time.

Table 14. Comparative study of RR, MMRR, IRRVQ and AMRR Algorithms with dynamic TQ

Algorithm	Average waiting time	Average Turnaround time
RR	96	131.4
MMRR	65.8	110
IRRVQ	82.8	118.2
AMRR	53.6	89

Figures 38, 39 below show the comparison of average waiting time and average turnaround time for the RR, MMRR, IRRVQ and AMRR algorithms.

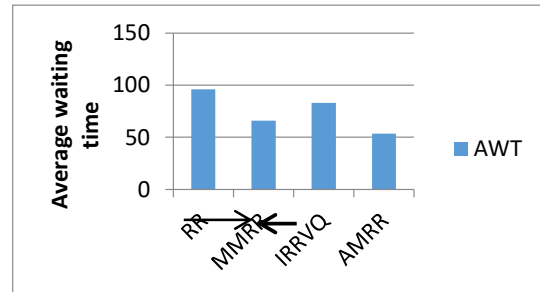


Figure 38: Graph for Average Waiting Time for RR, MMRR, IRRVQ and AMRR algorithms

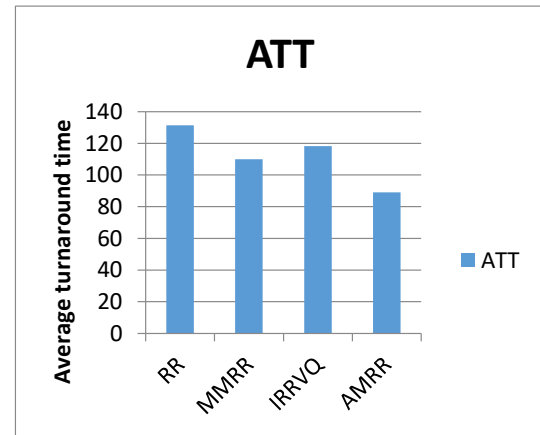


Figure 39: Graph for Average Turnaround Time for RR, MMRR, IRRVQ and AMRR algorithms

From the figures 38, 39 it is noticed that the algorithms MMRR, IRRVQ and AMRR are doing better than the standard Round Robin algorithm with respect to the average waiting time and the average turnaround time. By comparing the results of the other three algorithms, it is noticed that the MMRR algorithm reduces the average waiting time and the average turnaround time than the IRRVQ algorithm. However the AMRR algorithm is the best one of them.

4. Results analysis and amelioration

The TQ has two cases; static or dynamic. In the static TQ, each process has the same time slice where TQ may be either randomly chosen or calculated. The dynamic TQ is repeatedly calculated after each iteration.

In the static randomly chosen TQ; by comparing the algorithms RR, IRR, AAIRR and ERR, the AAIRR algorithm is the algorithm which gives the best results in random and decreasing cases. But in the case of increasing order it gives the same results as the ERR algorithm because of the benefit of AAIRR algorithm which is selecting the next short process to assign it to the processor which leads to minimize the average waiting time and average turnaround time does not exist.



In the static calculated TQ; by comparing the algorithms RR, Adaptive RR and RRRT, the algorithm RRRT gives better results than other algorithms. On the other hand in the dynamic TQ algorithms; by comparing the algorithms RR, MMRR, IRRVQ and AMRR, the AMRR algorithm performs better than other algorithms.

The results analysis shows that a combination of calculated the TQ in the static case may be ameliorated through the TQ equation of RRRT algorithm. This will be explained in the next section. Because the calculated TQ is more logic than randomly chosen TQ. So we recommended to use the Enriched Round Robin for minimizing the average waiting time and average turnaround time.

5. Enriched Round Robin algorithm

This algorithm improves the RRRT algorithm. If all the processes arrive at the same time to the ready queue, then all the processes are arranged in an ascending order according to their burst time. The TQ is calculated by $(\frac{3}{4} * \text{mean})$. If the remaining CPU burst time of the currently running process is less than the time quantum, the CPU is again allocated to the currently running process for the remaining CPU burst time. Otherwise, the process will be appended to the end of the ready queue. We proposed this algorithm that reduced the average waiting time and the average turnaround time in all cases which is published in [20] and proved that it improved the TQ that achieves stability in terms of the average waiting time and the average turnaround time.

6. Conclusion

Many algorithms are proposed to improve the efficiency of the standard Round Robin algorithm to minimize the average waiting time and average turnaround time. TQ is the most important issue in the performance of these algorithms; it can be static or dynamic. In static TQ, each process has the same time slice where TQ is either can be randomly chosen or calculated. Dynamic TQ is repeatedly calculated after each iteration.

From the above study, in static TQ algorithms it is noted that the algorithm AAIRR reduces the average waiting time and average turnaround time than other algorithms for the randomly chosen TQ. In the calculated TQ, we recommend that the Enriched Round Robin algorithm which gives the best results and achieving stability in terms of the average waiting time and the average turnaround time. On the other hand the dynamic TQ algorithms; the AMRR algorithm performs better than other algorithms.

7. References

- [1] Sukumar Babu Bandarupalli, Neelima Priyanka Nutulapati, P.Suresh Varma, "A Novel CPU Scheduling Algorithm-Preemptive & Non-Preemptive", International Journal of Modern Engineering Research (IJMER), Vol 2, Nov-Dec. 2012.
- [2] Rohith Roshan, K. Subra Rao, "Least-Mean Difference Round Robin (LMDRR) CPU Scheduling Algorithm", Journal of Theoretical and Applied Information Technology, Vol 88(1), June 2016.
- [3] Mohd Abdul Ahad, "Modifying Round Robin Algorithm for Process Scheduling using Dynamic Quantum Precision", Special Issue of International Journal of Computer Applications (0975 – 888) on Issues and Challenges in Networking, Intelligence and Computing Technologies – ICNICT 2012, November 2012.
- [4] Abbas Noon, Ali Kalakech and Seifedine Kadry, "A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average," International Journal of Computer Science, Vol 8(1), pp. 224-229, May 2011.
- [5] Sukumar Babu B., Neelima Priyanka N. and Sunil Kumar B., "Efficient Round Robin CPU Scheduling Algorithm," International Journal of Engineering Research and Development, Vol 4(9), Nov. 2012, p. 36-42.
- [6] Christopher McGuire, Jeonghwa Lee, "Comparisons of Improved Round Robin Algorithms", Proceedings of the World Congress on Engineering and Computer Science, Vol I, 2014.
- [7] A. Silberschatz, P. B. Galvin, and G. Gagne, "Operating System Concepts", (9th Edn. John Wiley and Sons Inc, 2013.
- [8] R.N.D.S.S. Kiran, Polinati Vinod Babu, and B.B. Murali Krishna, "Optimizing CPU Scheduling for Real Time Applications Using Mean-Difference Round Robin (MDRR) Algorithm", S.C. Satapathy et al. (eds.), ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of CSI - Volume I, Advances in Intelligent Systems and Computing 248 DOI: 10.1007/978-3-319-03107-1_78, © Springer International Publishing Switzerland 2014.
- [9] Anju Muraleedharan, Neenu Antony, R. Nandakumar, "Dynamic Time Slice Round Robin Scheduling Algorithm with Unknown Burst Time", Indian Journal of Science and Technology, Vol 9(8), pp.1-6, February 2016.
- [10] Imran Qureshi, "CPU Scheduling Algorithms: A Survey", Int. J. Advanced Networking and Applications, Vol 5(4), pp.1968-1973 (2014).
- [11] Manish Kumar Mishra, Abdul Kadir Khan, "An Improved Round Robin CPU Scheduling Algorithm", Journal of Global Research in Computer Science, Vol 3(6), pp. 64-69, June 2012.
- [12] Abdulrazaq Abdulrahim, Salisu Aliyu, Ahmad M Mustapha, Saleh E Abdullahi, "An Additional Improvement in Round Robin (AAAIRR) CPU Scheduling Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Vol 4(2), pp. 64-69, February 2014.
- [13] Jayanti Khatri, "An Enhanced Round Robin CPU Scheduling Algorithm", IOSR Journal of Computer Engineering (IOSR-JCE), Vol 18(4), pp. 20-24, Ver. II (Jul.-Aug. 2016).
- [14] Saroj Hiranwal, Dr. K.C. Roy "Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice", International Journal



- of Computer Science and Communication, Vol 2 (2), pp. 319-323, (July- December) 2011.
- [15] Arpita Sharma1, Mr. Gaurav Kakhani, "Analysis of Adaptive Round Robin Algorithm and Proposed Round Robin Remaining Time Algorithm", International Journal of Computer Science and Mobile Computing, Vol 4(12), pp.139-147, December 2015.
- [16] Rami J. Matarneh, "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of Now Running Processes", American J. of Applied Sciences Vol 6(10), 2009.
- [17] Sanjaya Kumar Panda, Sourav Kumar Bhoi, "An Effective Round Robin Algorithm using Min-Max Dispersion Measure", International Journal on Computer Science and Engineering (IJCSE), Vol 4(1), pp. 45-53, January 2012.
- [18] Manish Kumar Mishra, Dr. Faizur Rashid, "An Improved Round Robin CPU Scheduling Algorithm with Varying Time Quantum", International Journal of Computer Science, Engineering and Applications (IJCSEA), Vol 4(4), pp.1-8, August 2014.
- [19] Pallab Banerjee, Probal Banerjee, Shweta Sonali Dhal, "Comparative Performance Analysis of Average Max Round Robin Scheduling Algorithm (AMRR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static Time Quantum", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol 1(3), pp. 56-62, August 2012.
- [20] Kamal ElDahshan, Afaf Abd El-kader, Nermeen Ghazy, "Achieving Stability in the Round Robin Algorithm", International Journal of Computer Applications, Vol 172 (6), pp.15-20, August 2017.
- [21] Nermeen Ghazy, 2017 Enhancing Scheduling Jobs on Different Structures of Operating System. Master thesis. University of Al-Azhar.

Biographies

Prof. Kamal Abdelraouf ElDahshan



He is a professor of Computer Science and Information Systems at Al - Azhar University in Cairo, Egypt. An Egyptian national and graduate of Cairo University, he obtained his doctoral degree from the Université de Technologie de Compiègne in France, where he also taught for several years. During his extended stay in France, he also worked at the prestigious Institute National de Télécommunications in Paris. Professor ElDahshan's extensive international research, teaching, and consulting experiences have spanned four continents and include academic institutions as well as government and private organizations. He taught at Virginia Tech as a visiting professor; he was a Consultant to the Egyptian Cabinet Information and Decision Support Centre (IDSC); and he was a senior advisor to the Ministry of Education and Deputy Director of the National Technology Development Centre. Professor ElDahshan is a professional Fellow on Open Educational Resources as recognized by the United States Department of State and an Expert at ALECSO as recognized by the League of Arab States.

Dr. Afaf Abd El-Kader Abd EL-Hafiz



Is currently lecturer at University of Al-Azhar. She is doing her research work in Scheduling algorithms.

Nermeen Alaa Eldin Ghazy



She received her B.Sc. from Faculty of Science, AL-Azhar University at 2011. and Ms. Ghazy received M.Sc in scheduling algorithms at 2017.

